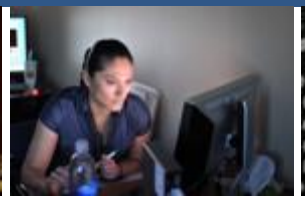
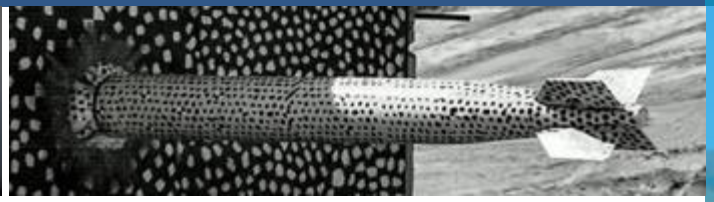
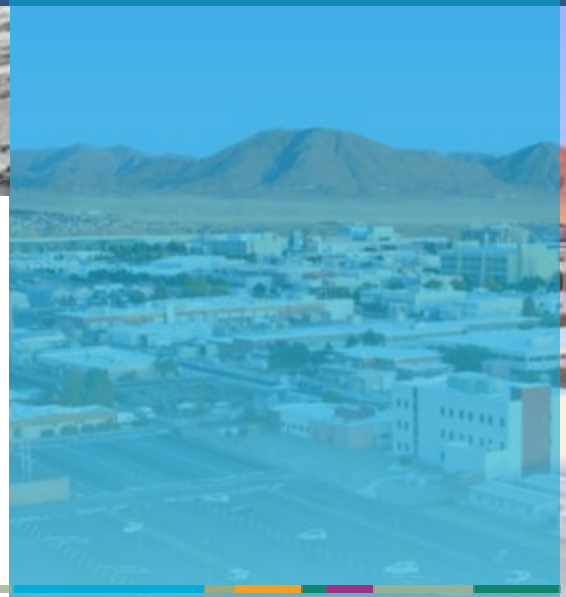
SAND2021-13382PE

Sandia National Laboratories

Data Science with Java







PRESENTED BY

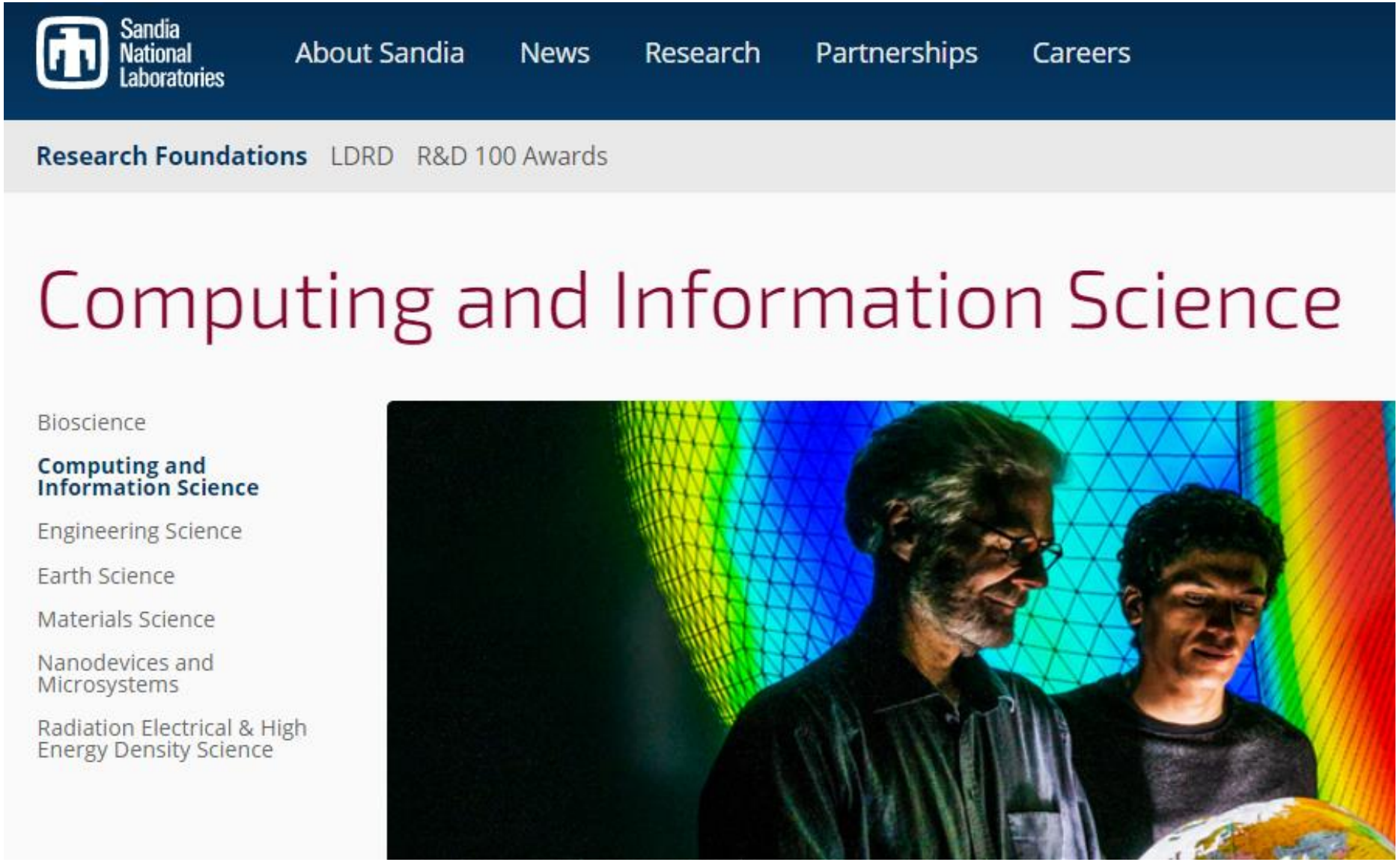
Michael Brzustowicz, PhD

Data Science and Cyber Analytics

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

https://www.energy.gov/stem-rising

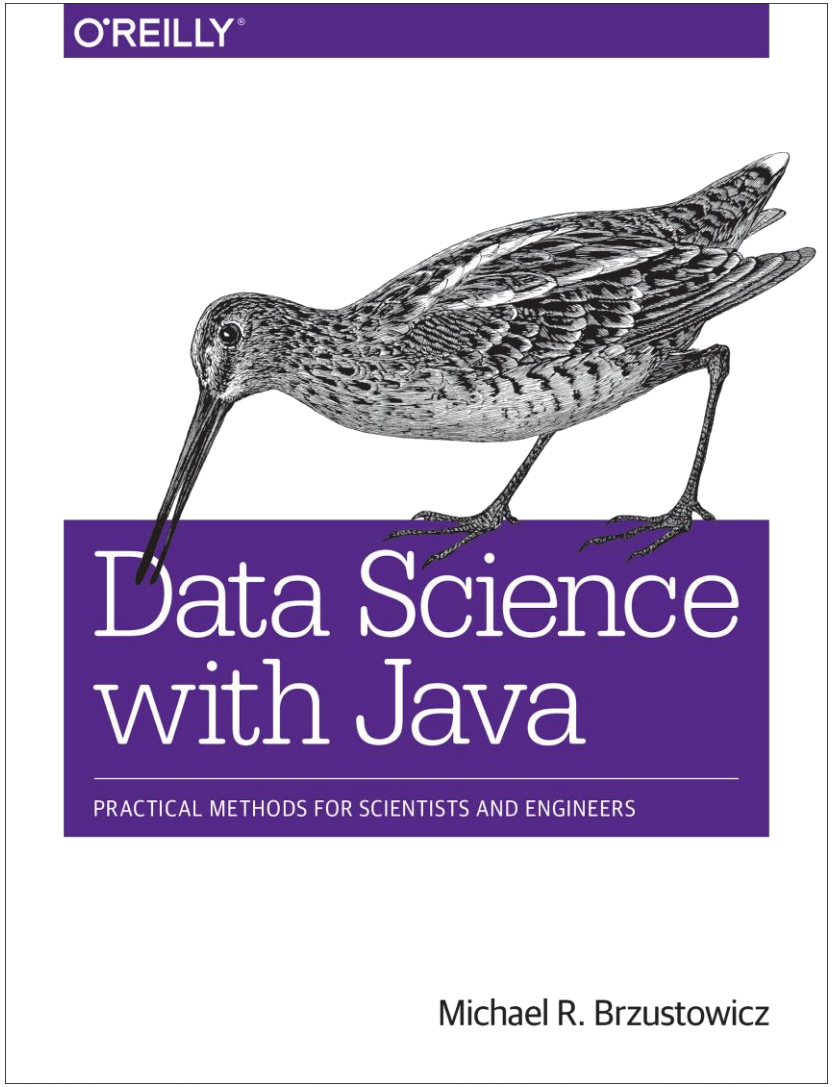


https://www.sandia.gov/research/research_foundations/computing_information_science.html

I wrote this book →

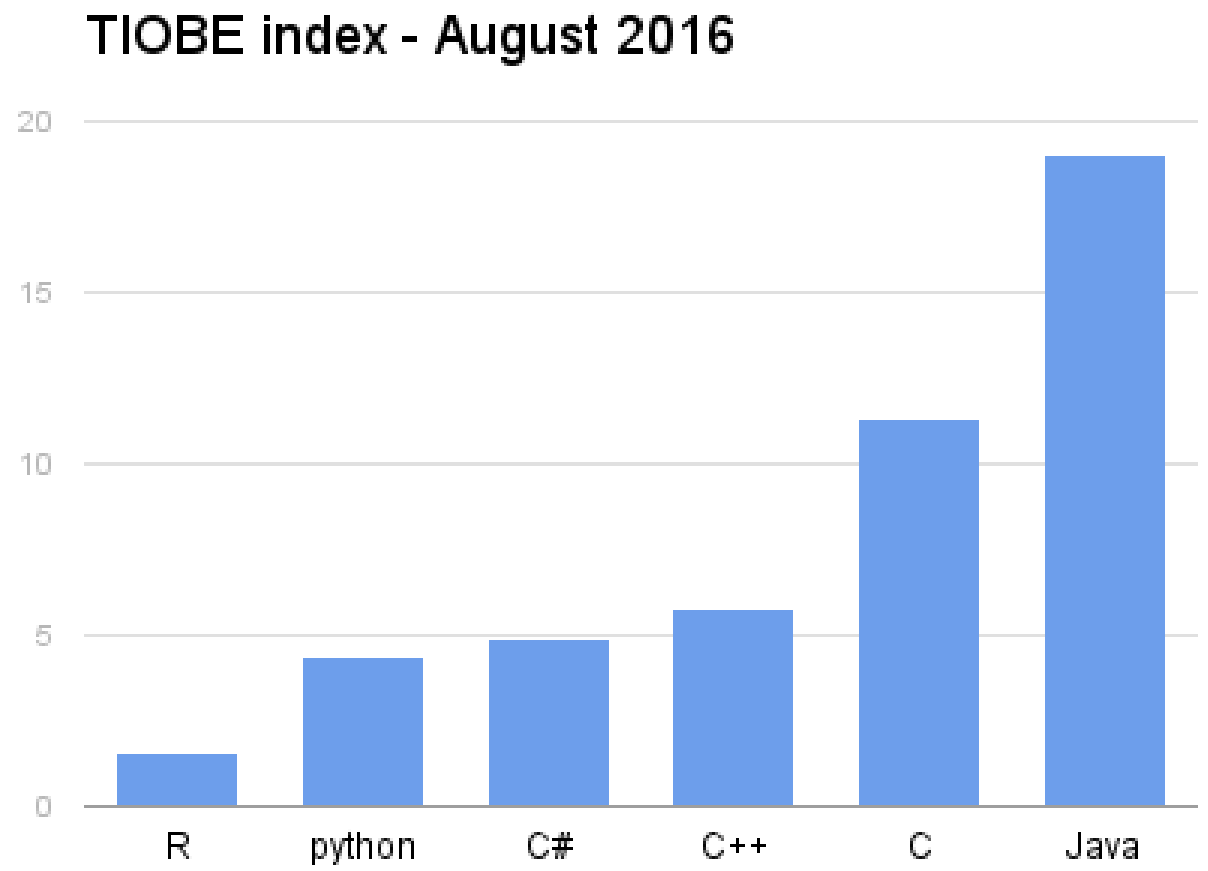
I am giving a lot of thought to what version 2 will be like.

Code in book is at:
https://github.com/oreillymedia/Data_Science_with_Java



There’s approximately 10 million Java developers world wide.

https://en.wikipedia.org/wiki/Java_(programming_language)



Java had the highest percent increase over 2015
Java has been #1 (on average) for ~ 15 years

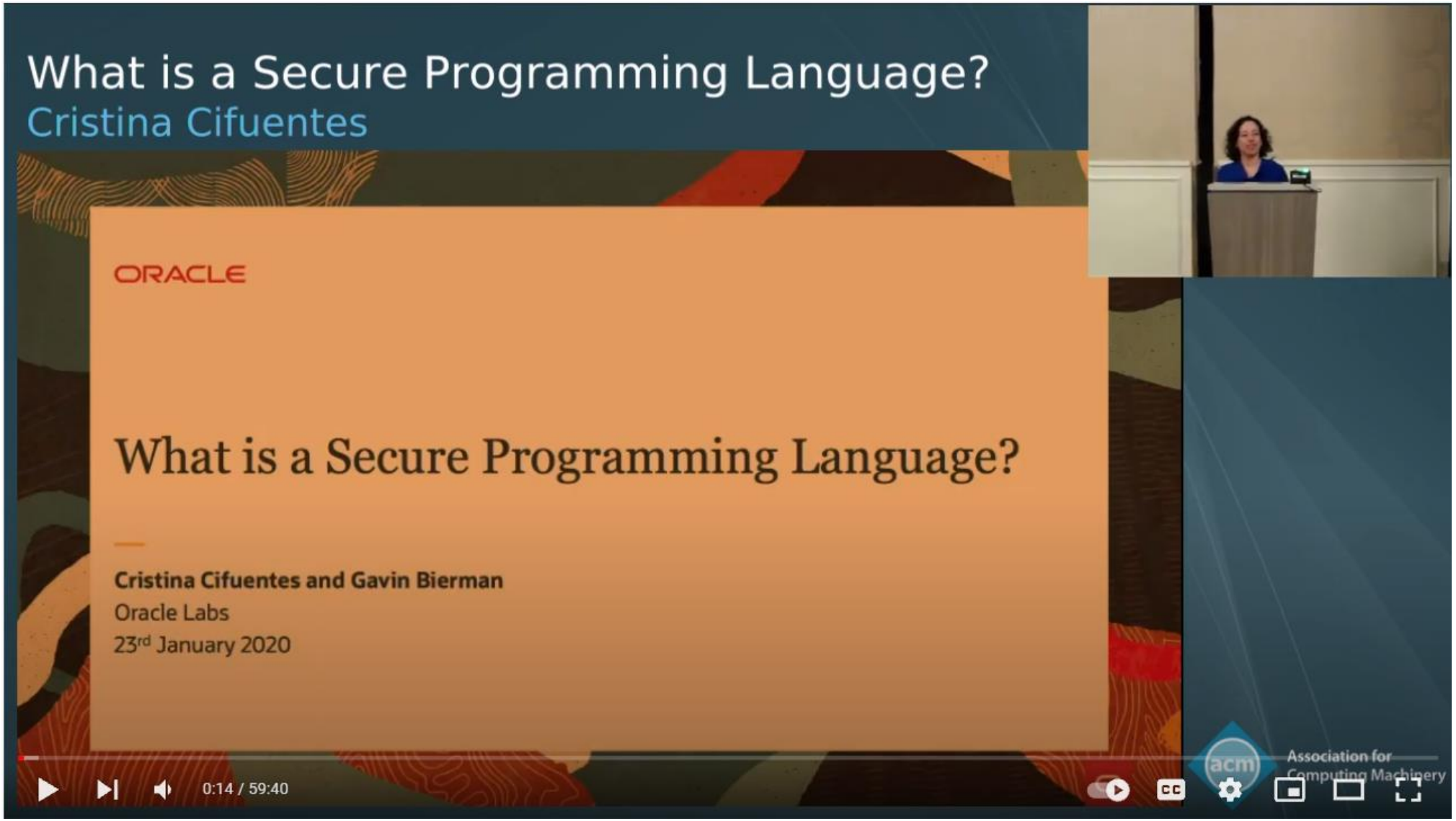
In 2022 Python is leading the pack …

Oct 2021	Oct 2020	Change	Programming Language	Ratings	Change
1	3	↑	Python	11.27%	-0.00%
2	1	↓	C	11.16%	-5.79%
3	2	↓	Java	10.46%	-2.11%
4	4		C++	7.50%	+0.57%
5	5		C#	5.26%	+1.10%
6	6		Visual Basic	5.24%	+1.27%
7	7		JavaScript	2.19%	+0.05%
8	10	↑	SQL	2.17%	+0.61%
9	8	↓	PHP	2.10%	+0.01%
10	17	⇈	Assembly language	2.06%	+0.99%
11	19	⇈	Classic Visual Basic	1.83%	+1.06%
12	14	↑	Go	1.28%	+0.13%
13	15	↑	MATLAB	1.20%	+0.08%
14	9	⇊	R	1.20%	-0.79%

https://www.tiobe.com/tiobe-index/

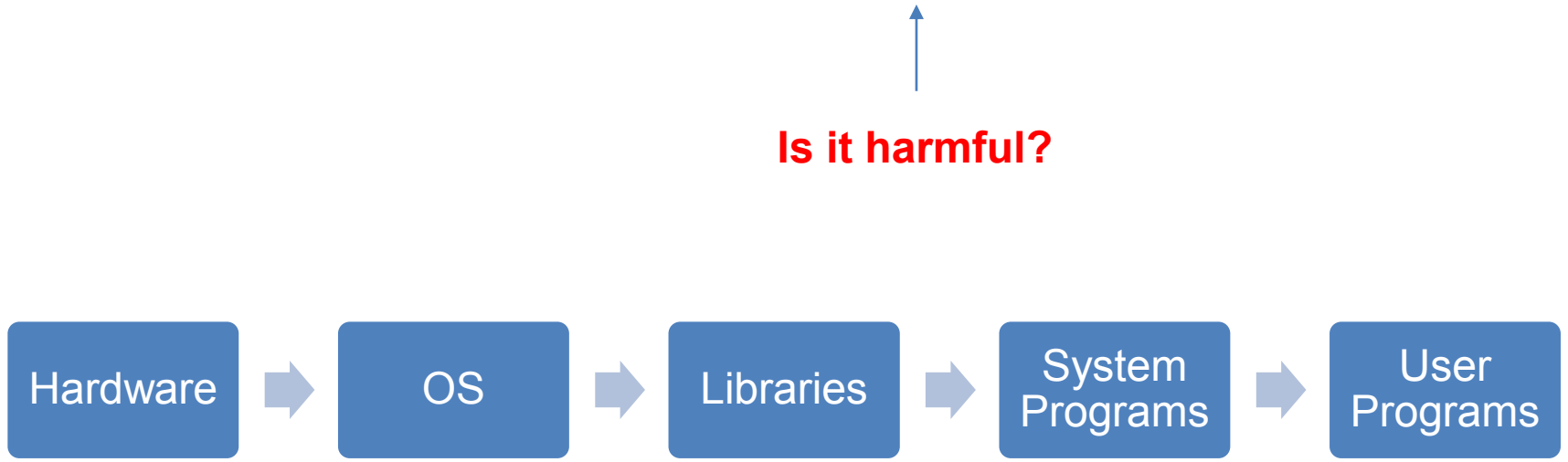
Why Data Science with Java?

Java is solid and secure.
A lot of infrastructure is coded in Java.
Why not add Data Science to the mix?

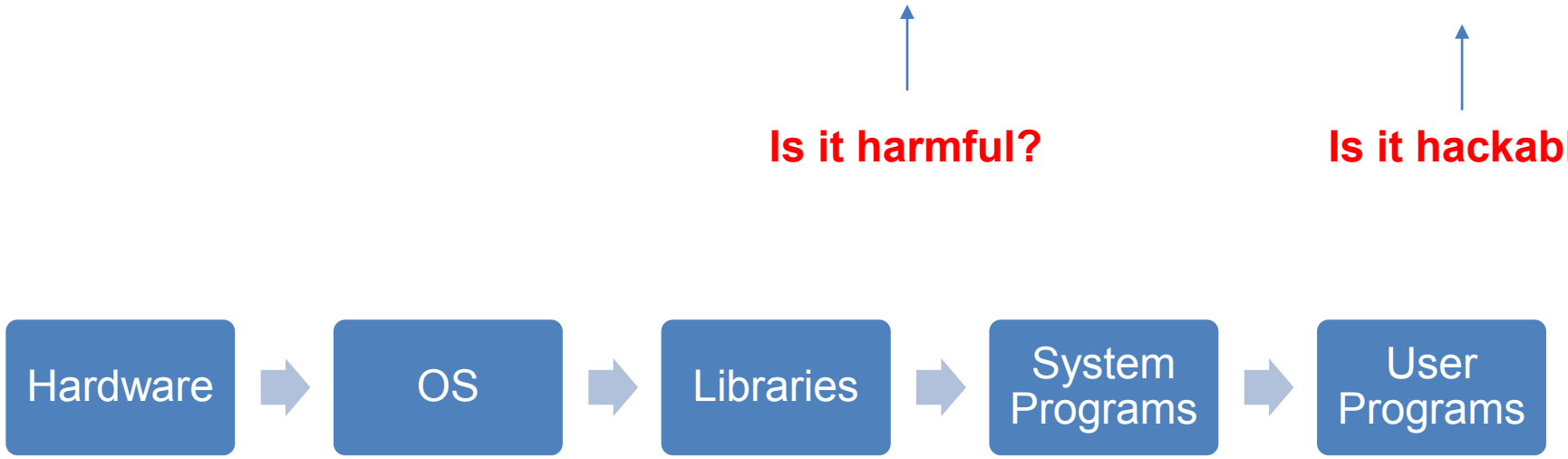

What is a Secure Programming Language?
Cristina Cifuentes
ORACLE
What is a Secure Programming Language?
Cristina Cifuentes and Gavin Bierman
Oracle Labs
23rd January 2020
0:14 / 59:40
acm
Association for Computing Machinery

My Thoughts on AI Safety and Security

My Thoughts on AI Safety and Security



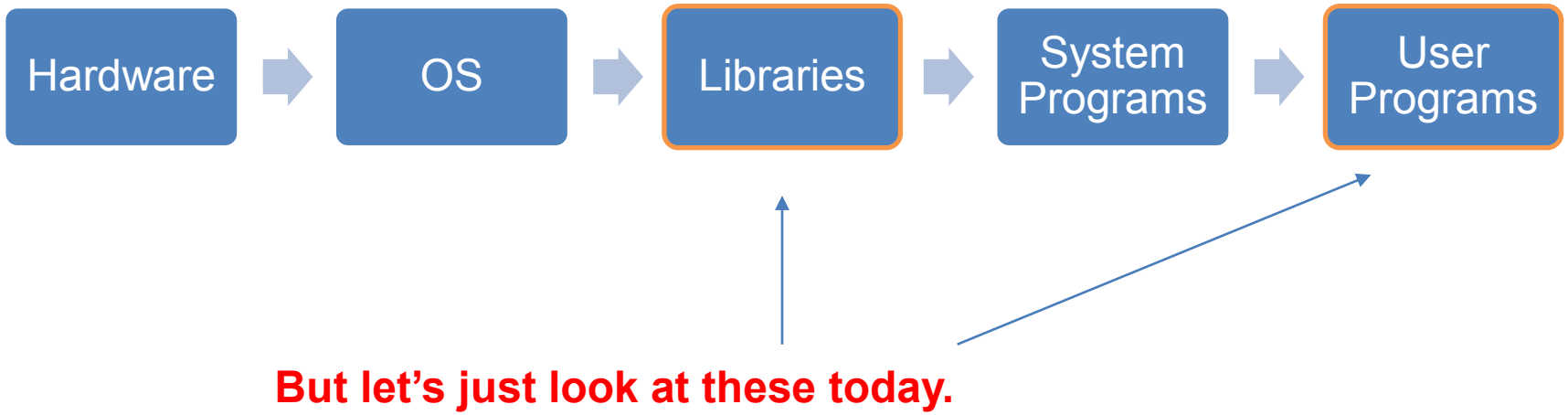
My Thoughts on AI Safety and Security



My Thoughts on AI Safety and Security

Any one of these is a potential point of failure.

My Thoughts on AI Safety and Security



Why not Python?

Things that really bug me about Python:

1) white space
    - broken code tabs vs. space going between editors
    - broken code when commenting out blocks … temporary tabs
2) weak types
    - forgetting to cast input string as int or float (this always gets me)
    - makes documentation almost useless, i.e. not knowing input or return types
    - how many different ways to represent Boolean?
3) library
    - bloat (too many choices sometimes)
    - deprecated libraries / abandoned projects
    - version mismatch especially if it involves numpy
    - the solution seems to be venv (python has a pvm too now!)
    - just who's in charge anyway? pip or apt-get / yum or …
4) docs
    - python docs are usually not so great (with sklearn being an exception)
5) object orientation
    - no true encapsulation
    - __do_not_look_at_this()

- As an example, would you rather read CSV parser docs in Python or Java?

pandas.read_csv

`pandas.`**`read_csv`**(*filepath_or_buffer, sep=', ', delimiter=None, header='infer', names=None, index_col=None, usecols=None, squeeze=False, prefix=None, mangle_dupe_cols=True, dtype=None, engine=None, converters=None, true_values=None, false_values=None, skipinitialspace=False, skiprows=None, skipfooter=0, nrows=None, na_values=None, keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True, parse_dates=False, infer_datetime_format=False, keep_date_col=False, date_parser=None, dayfirst=False, iterator=False, chunksize=None, compression='infer', thousands=None, decimal=b'.', lineterminator=None, quotechar='"', quoting=0, doublequote=True, escapechar=None, comment=None, encoding=None, dialect=None, tupleize_cols=None, error_bad_lines=True, warn_bad_lines=True, delim_whitespace=False, low_memory=True, memory_map=False, float_precision=None*) ¶ [source]

Read a comma-separated values (csv) file into DataFrame.

Also supports optionally iterating or breaking of the file into chunks.

Additional help can be found in the online docs for IO Tools.

filepath_or_buffer : *str, path object, or file-like object*

Any valid string path is acceptable. The string could be a URL. Valid URL schemes include http, ftp, s3, and file. For file URLs, a host is expected. A local file could be: file://localhost/path/to/table.csv.

If you want to pass in a path object, pandas accepts either `pathlib.Path` or `py._path.local.LocalPath`.

By file-like object, we refer to objects with a `read()` method, such as a file handler (e.g. via builtin `open` function) or `StringIO`.

static CSVFormat	**INFORMIX_UNLOAD** Default Informix CSV UNLOAD format used by the `UNLOAD TO file_name` operation.
static CSVFormat	**INFORMIX_UNLOAD_CSV** Default Informix CSV UNLOAD format used by the `UNLOAD TO file_name` operation (escaping is disabled.)
static CSVFormat	**MONGODB_CSV** Default MongoDB CSV format used by the `mongoexport` operation.
static CSVFormat	**MONGODB_TSV** Default MongoDB TSV format used by the `mongoexport` operation.
static CSVFormat	**MYSQL** Default MySQL format used by the `SELECT INTO OUTFILE` and `LOAD DATA INFILE` operations.
static CSVFormat	**ORACLE** Default Oracle format used by the SQL*Loader utility.
static CSVFormat	**POSTGRESQL_CSV** Default PostgreSQL CSV format used by the `COPY` operation.
static CSVFormat	**POSTGRESQL_TEXT** Default PostgreSQL text format used by the `COPY` operation.
static CSVFormat	**RFC4180** Comma separated format as defined by **RFC 4180**.
static CSVFormat	**TDF** Tab-delimited format.

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
boolean	**equals**(**Object** obj)
String	**format**(**Object**... values) Formats the specified values.
boolean	**getAllowMissingColumnNames**() Specifies whether missing column names are allowed when parsing the header line.

Java Docs are universal and uniform

Package org.hipparchus.linear

Interface RealMatrix

All Superinterfaces:

AnyMatrix

All Known Subinterfaces:

SparseRealMatrix

All Known Implementing Classes:

AbstractRealMatrix, Array2DRowRealMatrix, BlockRealMatrix, DiagonalMatrix, OpenMapRealMatrix

---

```
public interface RealMatrix
extends AnyMatrix
```

Interface defining a real-valued matrix with basic algebraic operations.

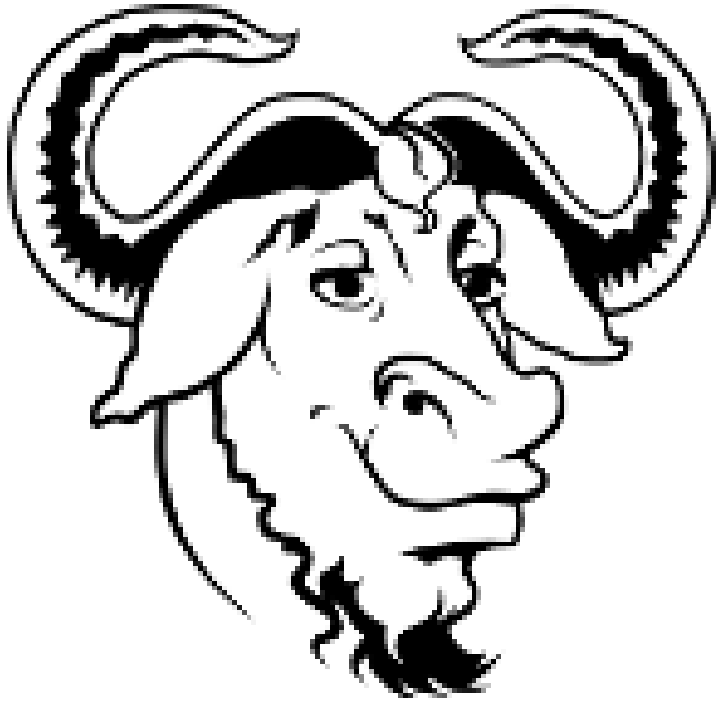
Matrix element indexing is 0-based -- e.g., `getEntry(0, 0)` returns the element in the first row, first column of the matrix.

Method Summary

All Methods | Instance Methods | Abstract Methods | Default Methods

Modifier and Type	Method	Description
RealMatrix	add(RealMatrix m)	Returns the sum of this and m.
void	addToEntry(int row, int column, double increment)	Adds (in place) the specified value to the specified entry of this matrix.
RealMatrix	copy()	Returns a (deep) copy of this.
void	copySubMatrix(int[] selectedRows, int[] selectedColumns, double[][] destination)	Copy a submatrix.
void	copySubMatrix(int startRow, int endRow, int startColumn, int endColumn, double[][] destination)	Copy a submatrix.
RealMatrix	createMatrix(int rowDimension, int columnDimension)	Create a new RealMatrix of the same type as the instance with the

Is Apache the new GNU?



https://projects.apache.org/statistics.html

Projects Statistics

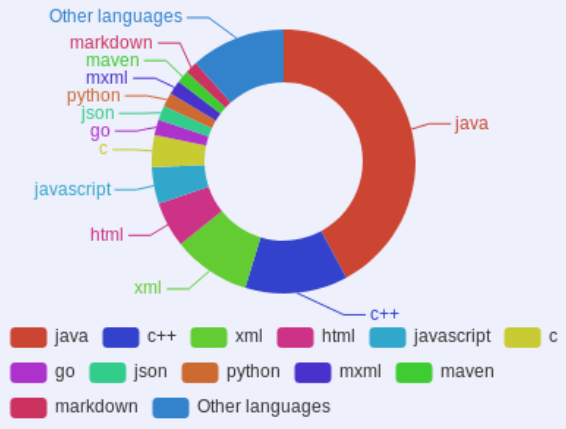
Home | Committees | Projects | Releases | **Statistics** | Timelines | Search...

Statistics

Codebase statistics

Language breakdown across the ASF:

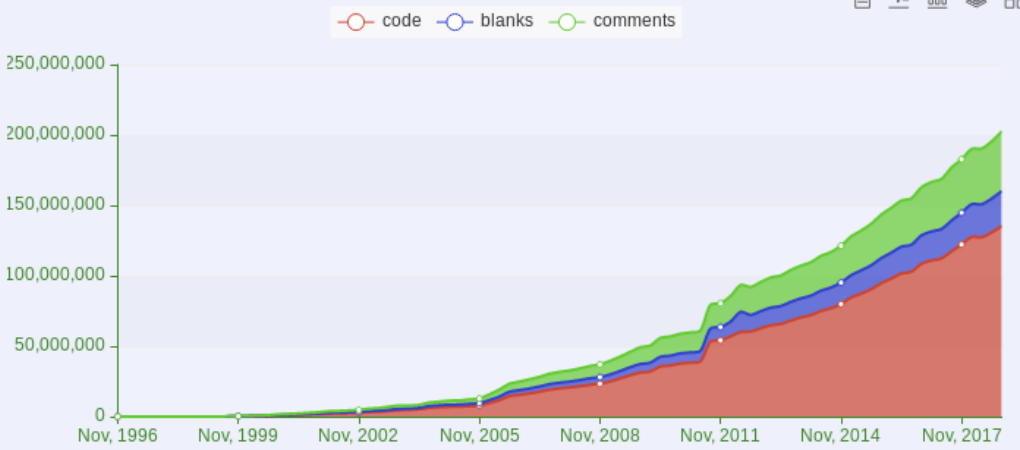
This chart shows the languages used by all projects at the ASF, sorted by the number of lines per language (excluding comments and blanks)



Data courtesy of Snoot.io

Evolution of codebase over time:

This chart shows the lines of code, blanks and comments over time at the ASF (see cloc documentation for details).

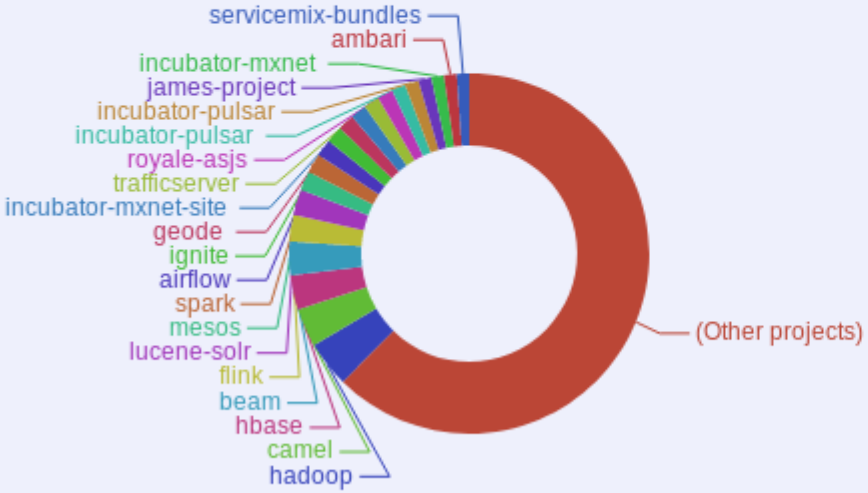


Data courtesy of Snoot.io

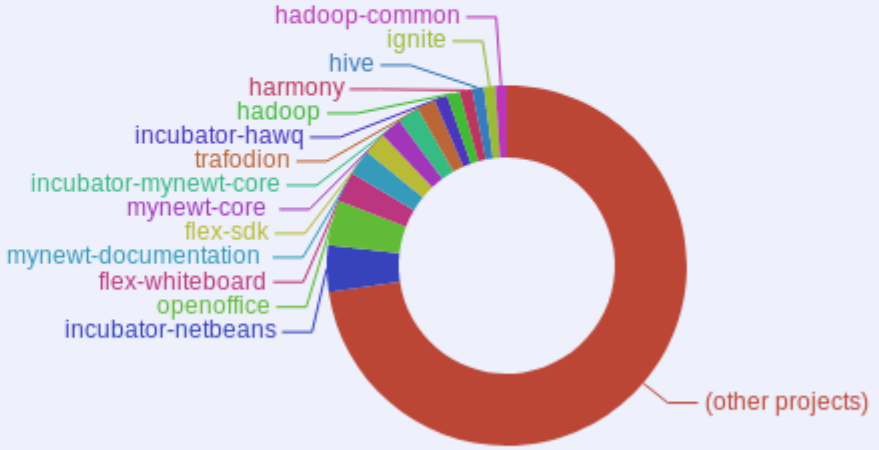
https://projects.apache.org/statistics.html

Largest/Busiest repositories

Repositories by number of commits, past year:



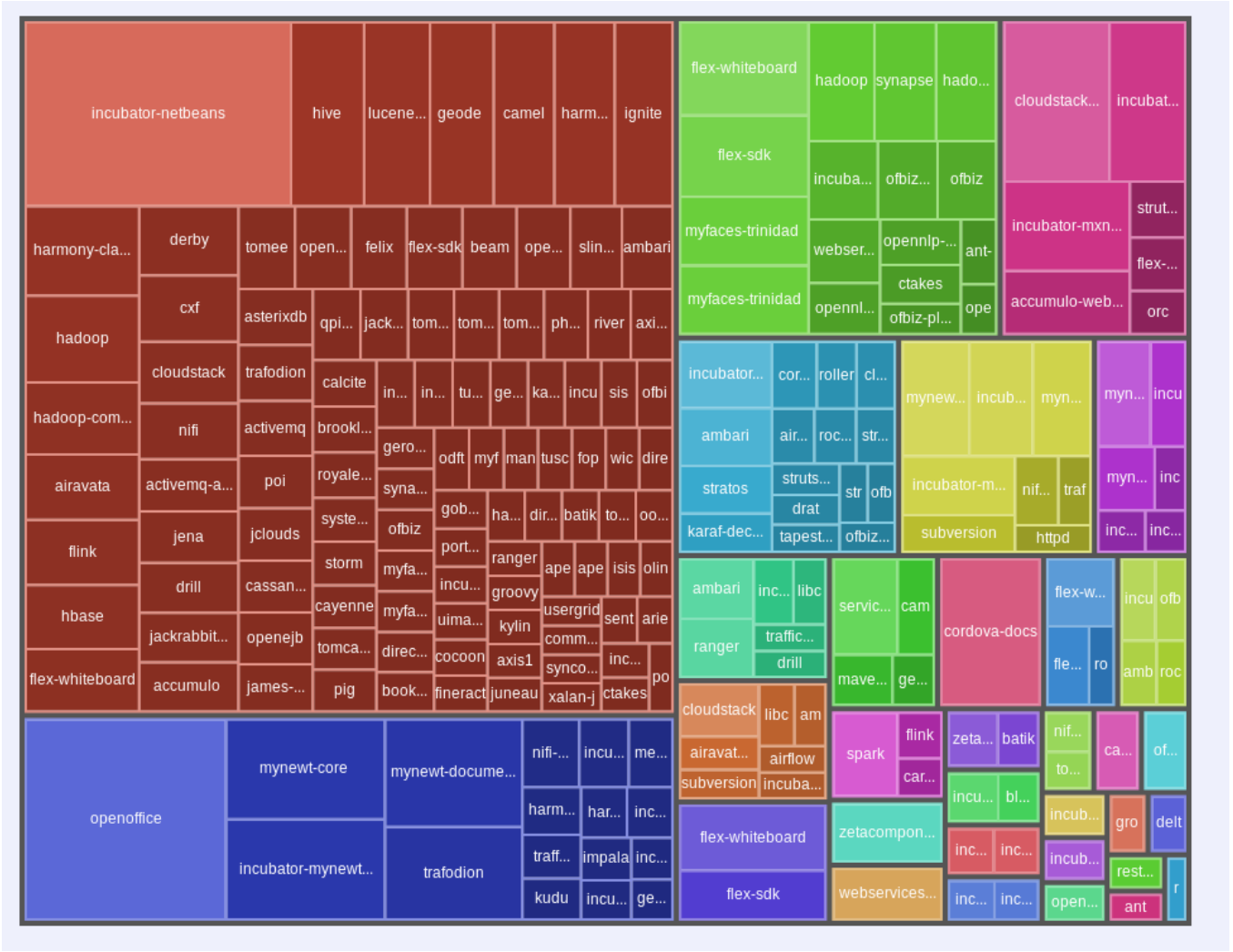
Repositories by Lines of Code:

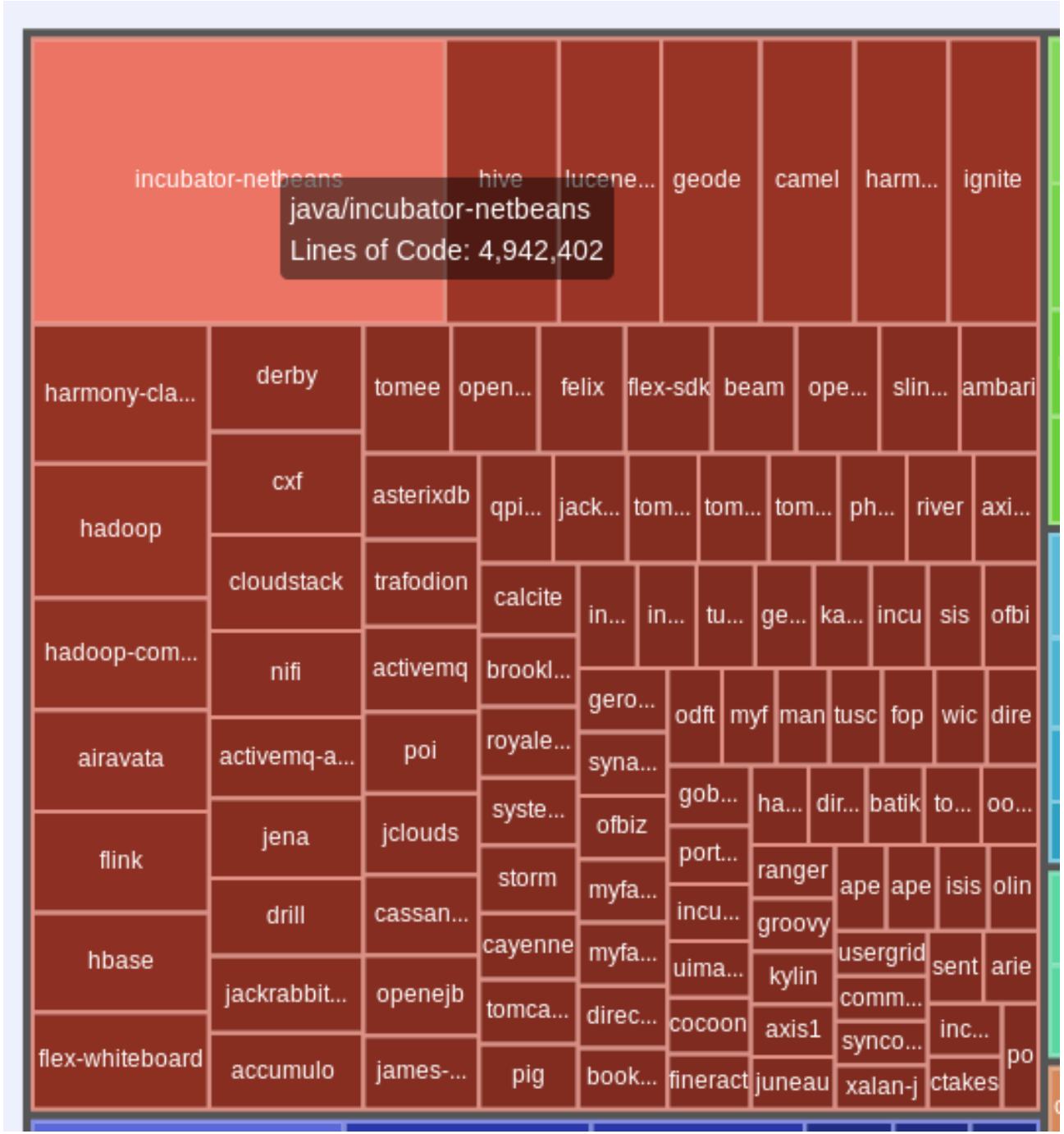


https://projects.apache.org/statistics.html

Apache Projects broken down by source

…



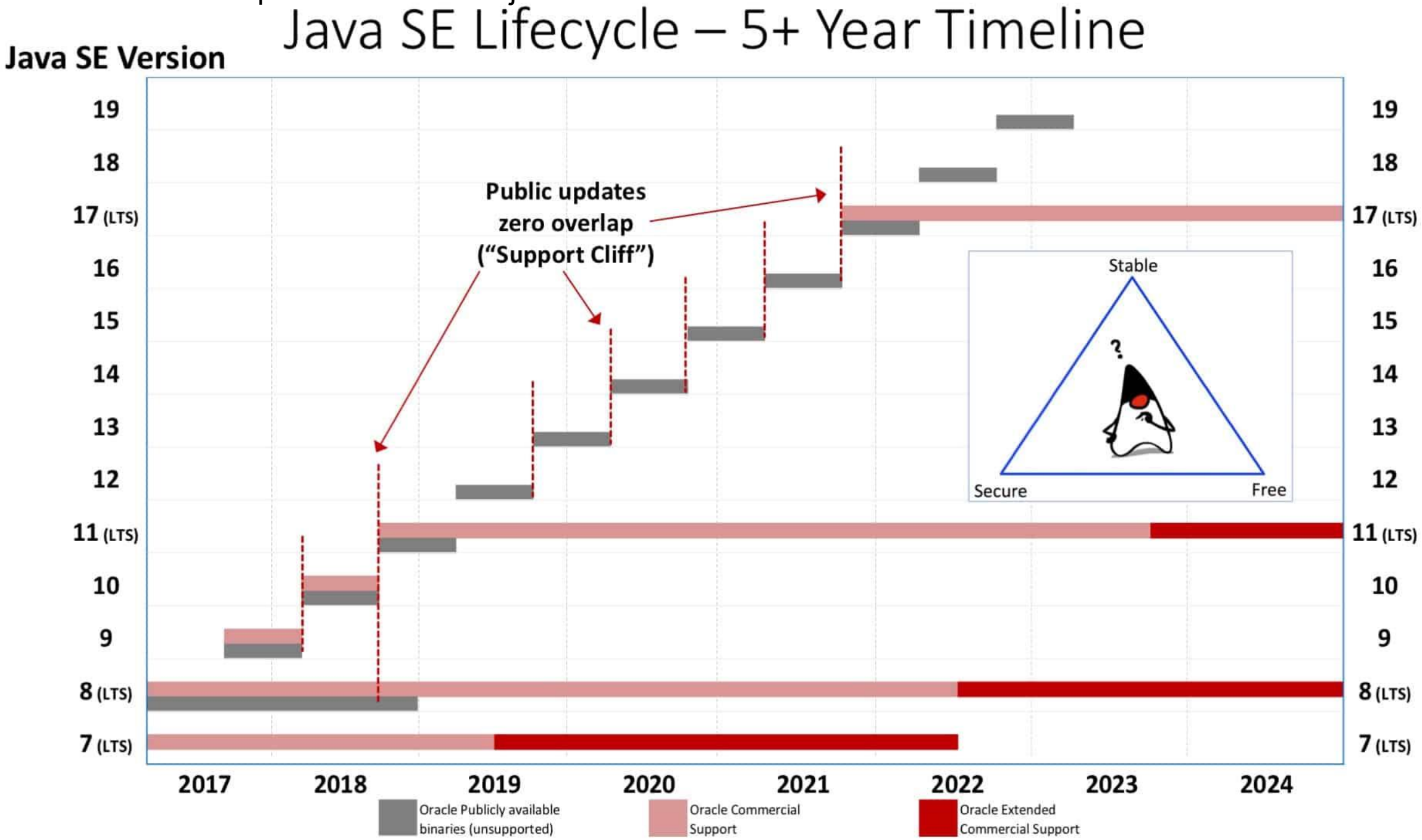


https://projects.apache.org/statistics.html

… has a lot of Java.

What’s going on with Java?

https://www.azul.com/java-stable-secure-free-choose-two-three/



Red Hat OpenJDK 11 Advice

OpenJDK 11 included >= RHEL 7.6

Update apps from JDK8 → OpenJDK 11

Red Hat will support OpenJDK 11 till at least 2024

https://access.redhat.com/articles/3409141

OpenJDK is the new JDK.

A History of Mathematical Java.

Keep in mind ...

From the beginning (1995), Java was no C or FORTRAN.

Java was comparatively slow.

Because ...
Java was originally created to address SECURE NETWORKING.

Java Math class (standard algorithms in C with plans to rewrite in Java)

e.g. cos, tanh, abs, etc.

NIST JAMA (https://math.nist.gov/javanumerics/jama/)

Matrix, Matrix ops, SVD, eigenvalue
commits: 1998, 2000, 2003, 2005, 2012
see https://math.nist.gov/javanumerics/jama/ChangeLog
JAMA was supposed to become part of the JDK? [Didn't happen]

Apache Commons Math (https://commons.apache.org/proper/commons-math/)

Linear Algebra, Sparse Linear Algebra, Statistics (includes streaming), Clustering, Optimization, Neural Networks (kind of), many other things
2007 - current

Hipparchus (https://www.hipparchus.org/)

Re-factor Apache Commons Math into modules.
Fold back to ACM? Or become a new Apache TLP?
2016 - current

javax.vecmath

(https://docs.oracle.com/cd/E17802_01/j2se/javase/technologies/desktop/java3d/forDevelopers/j3dapi/javax/vecmath/package-summary.html)
GMatrix, Gvector [G is for "general", package was for Java 3D graphics]
20xx – not maintained

javax.vecmath

(https://docs.oracle.com/cd/E17802_01/j2se/javase/technologies/desktop/java3d/forDevelopers/j3dapi/javax/vecmath/package-summary.html)

GMatrix, Gvector [G is for “general”, package was for Java 3D graphics]

20xx – not maintained

Algorithm Foundry (https://github.com/algorithmfoundry/Foundry)

formerly “Cognitive Foundry” as a Sandia project

Weka (https://www.cs.waikato.ac.nz/~ml/weka/)

more of an app than libs

DL4J (https://deeplearning4j.org/)

commercially backed

Tribuo (tribuo.org) – Machine Learning / Data Science Library from Oracle Labs

Java Vector (Coming in Java 17?) – native Vector Class is coming (no … it’s not the OLD, deprecated ‘Vector Class’, it’s a new one, with CPU optimization.

Ideal Mathematical Java

Ideally, stable mathematical algorithms should be part of Java (JDK)

But considering failure of javax.vecmath and java.stats, probably will never happen.
Too bad … Java could be the “Language of AI”

Ideally, stable mathematical algorithms should be part of Java (JDK)

But considering failure of javax.vecmath and java.stats, probably will never happen.
Too bad … Java could be the "Language of AI"

THINGS HAVE CHANGED SINCE I LAST GAVE THIS TALK IN 2019!

Looks like we are going to get some native Vector Classes and Linear Algebra?
But who knows … and until then …

Ideally, stable mathematical algorithms should be part of Java (JDK)

But considering failure of javax.vecmath and java.stats, probably will never happen.
Too bad … Java could be the "Language of AI"

Instead, stable mathematical algorithms should be Apache TLP

We already have great code in ACM and Hipparchus (goal is TLP with modules)

Machine Learning needs a lot of work …
Modern NN, Ensemble Methods, Text (NLU), Learning Metrics

Caution: don't just cut and paste old code!

Ideally, stable mathematical algorithms should be part of Java (JDK)

But considering failure of javax.vecmath and java.stats, probably will never happen.
Too bad … Java could be the "Language of AI"

Instead, stable mathematical algorithms should be Apache TLP

We already have great code in ACM and Hipparchus (goal is TLP with modules)

Machine Learning needs a lot of work …
Modern NN, Ensemble Methods, Text (NLU), Learning Metrics

Caution: don't just cut and paste old code!

CALL TO ACTION!!!

What do we really need for Data Science?

Data Science Library

1) IO – Input parsing and checking, output formatting and plotting

2) Linear Algebra – Matrix operations and decomposition

3) Statistics – Descriptive stats, random number distributions and information theory

4) Data Operations – Transforming text, reducing dimension, scaling, encoding, partitioning

5) Learning – algorithms and metrics

Data Science Library

1) IO – Input parsing and checking, output formatting and plotting [Apache Commons, JavaFX]

2) Linear Algebra – Matrix operations and decomposition

3) Statistics – Descriptive stats, random number distributions and information theory

4) Data Operations – Transforming text, reducing dimension, scaling, encoding, partitioning

5) Learning – algorithms and metrics

Data Science Library

1) IO – Input parsing and checking, output formatting and plotting [Apache Commons, JavaFX]

2) Linear Algebra – Matrix operations and decomposition [Apache Commons, Hipparchus]

3) Statistics – Descriptive stats, random number distributions and information theory

4) Data Operations – Transforming text, reducing dimension, scaling, encoding, partitioning

5) Learning – algorithms and metrics

Data Science Library

1) IO – Input parsing and checking, output formatting and plotting [Apache Commons, JavaFX]

2) Linear Algebra – Matrix operations and decomposition [Apache Commons, Hipparchus]

3) Statistics – Descriptive stats, random number distributions and information theory [ACM]

4) Data Operations – Transforming text, reducing dimension, scaling, encoding, partitioning

5) Learning – algorithms and metrics

Data Science Library

1) IO – Input parsing and checking, output formatting and plotting [Apache Commons, JavaFX]

2) Linear Algebra – Matrix operations and decomposition [Apache Commons, Hipparchus]

3) Statistics – Descriptive stats, random number distributions and information theory [ACM]

4) Data Operations – Transforming text, reducing dimension, scaling, encoding, partitioning

5) Learning – algorithms and metrics [Clustering in Apache Commons]

Data Science Library

1) IO – Input parsing and checking, output formatting and plotting [Apache Commons, JavaFX]

2) Linear Algebra – Matrix operations and decomposition [Apache Commons, Hipparchus]

3) Statistics – Descriptive stats, random number distributions and information theory [ACM]

4) Data Operations – Transforming text, reducing dimension, scaling, encoding, partitioning

5) Learning – algorithms and metrics [Clustering in Apache Commons]

We need …

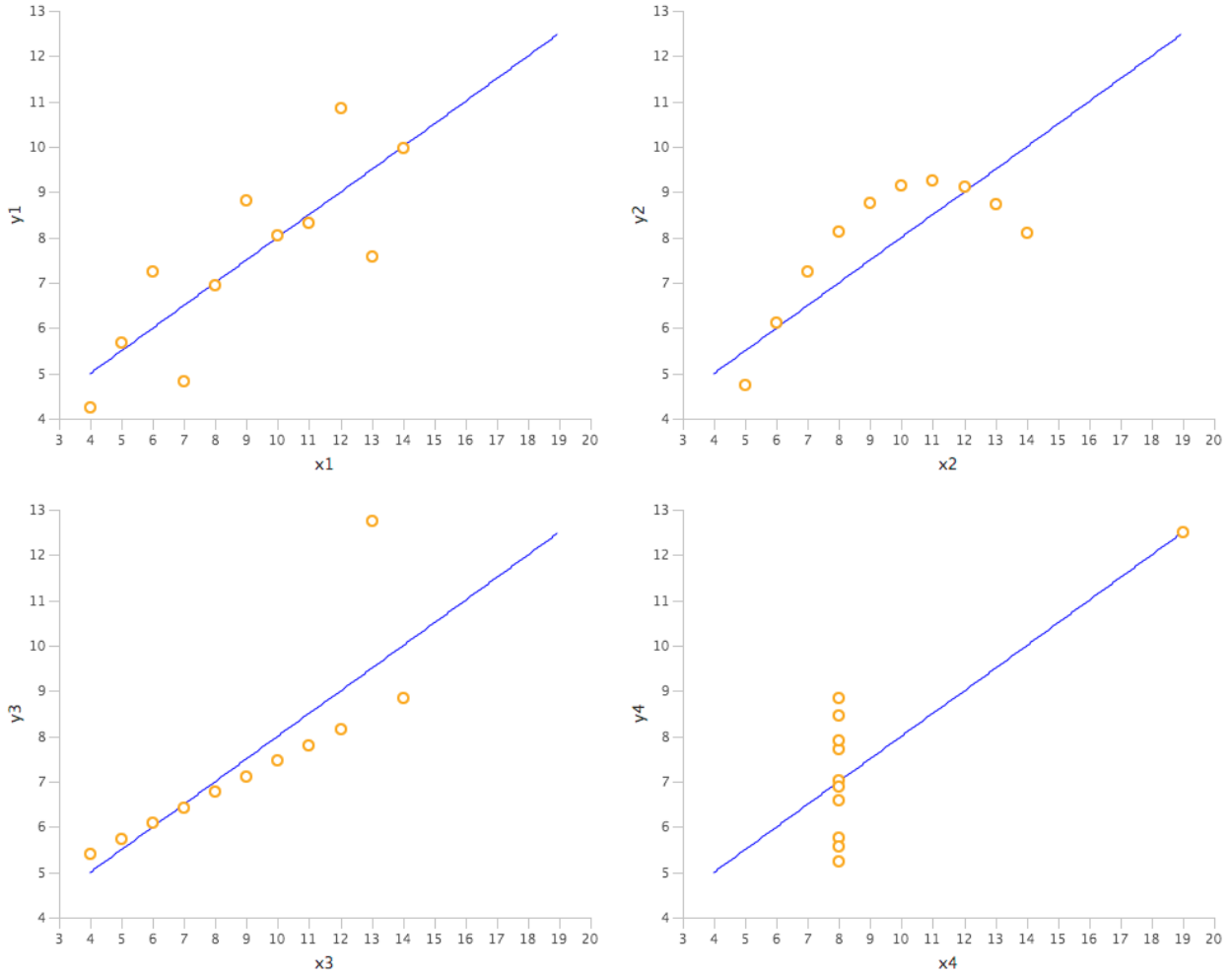
IO – Combine CSV, TSV, JSON
IO – JavaFX not in the OpenJDK? Could make it easier to use also.
Data Operations – D.N.E.
Learning – include generalized EM algorithm in addition to generalized optimizer?
Learning – fold linear and non-linear regression into here
Learning – add DNN, RF, SVM, KNN etc...

Plotting Example

Anscombe’s Quartet

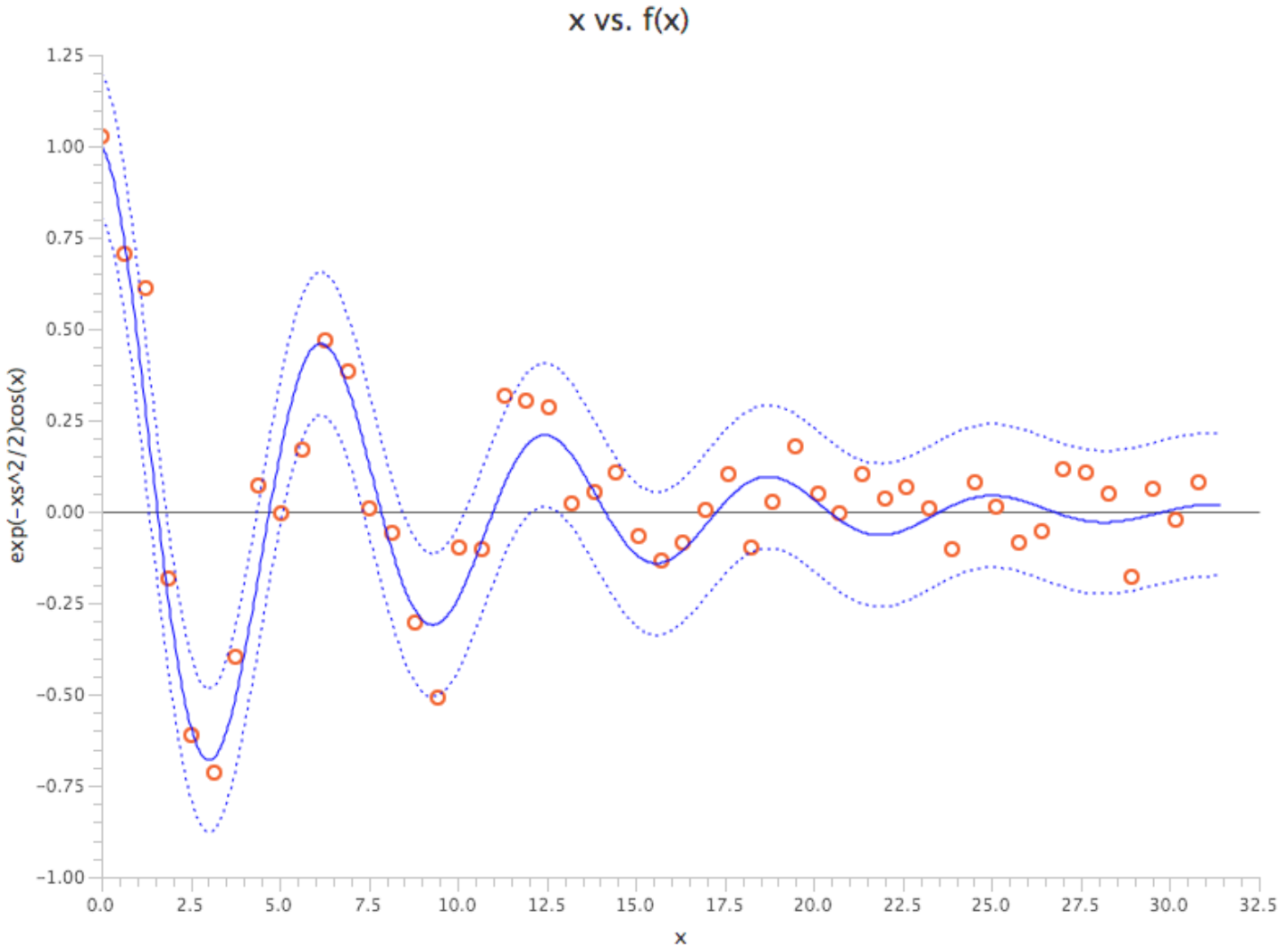
Data from Wikipedia

Plots made with JavaFX



Random Data with 3 Sigma Bands

Random Data
Plots made with JavaFX



JavaFX

Official Java graphics package

Used to be separate package

Then it was included in JDK8!

Now it is a separate package … again …

And it doesn’t work so great anymore :(

Has lots of features and great for building GUI apps.

Could be easier to use :(

JavaFX

Official Java graphics package

Used to be separate package

Then it was included in JDK8!

Now it is a separate package … again …

And it doesn't work so great anymore :(

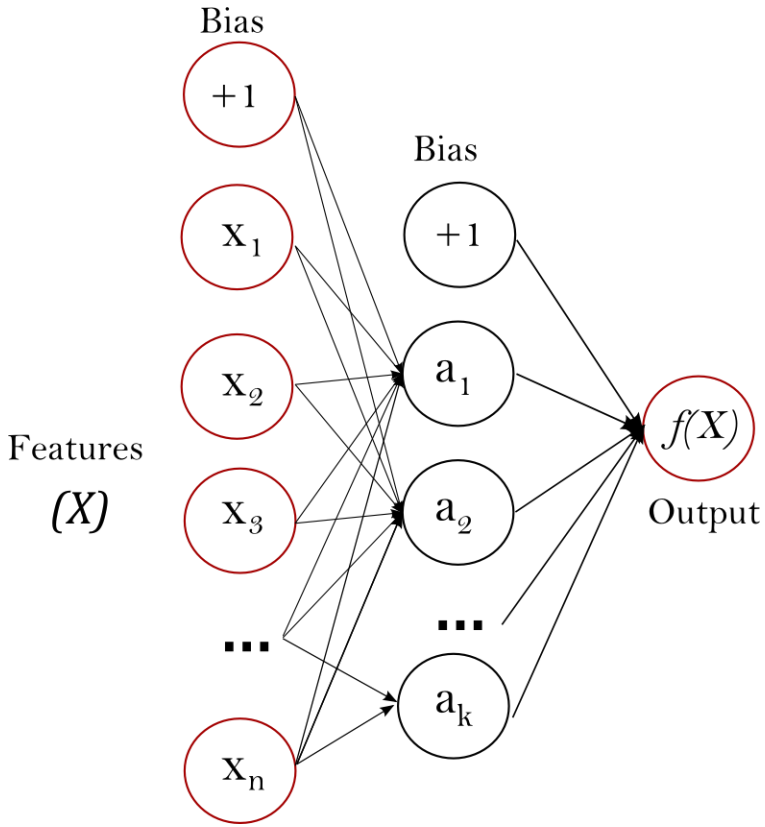
Has lots of features and great for building GUI apps.

Could be easier to use :(

```
//TODO make it like this without extending Application class
Plot plot = new ScatterPlot(xData, yData)
plot.render()
```

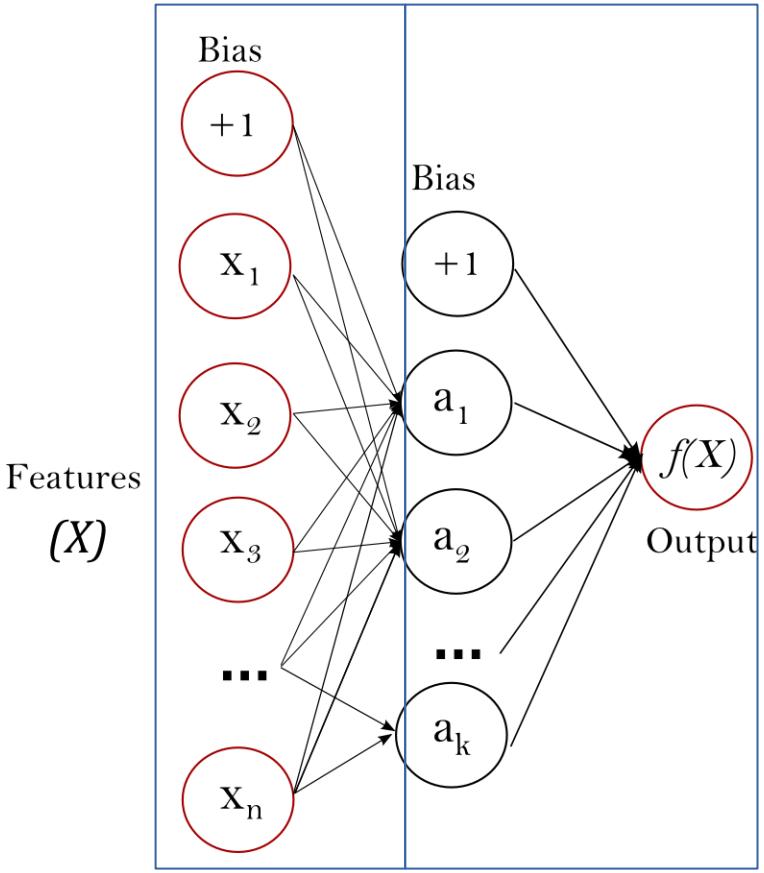
Learning Example: MLP (Neural Network)

Multi-layer Perceptron



https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Multi-layer Perceptron

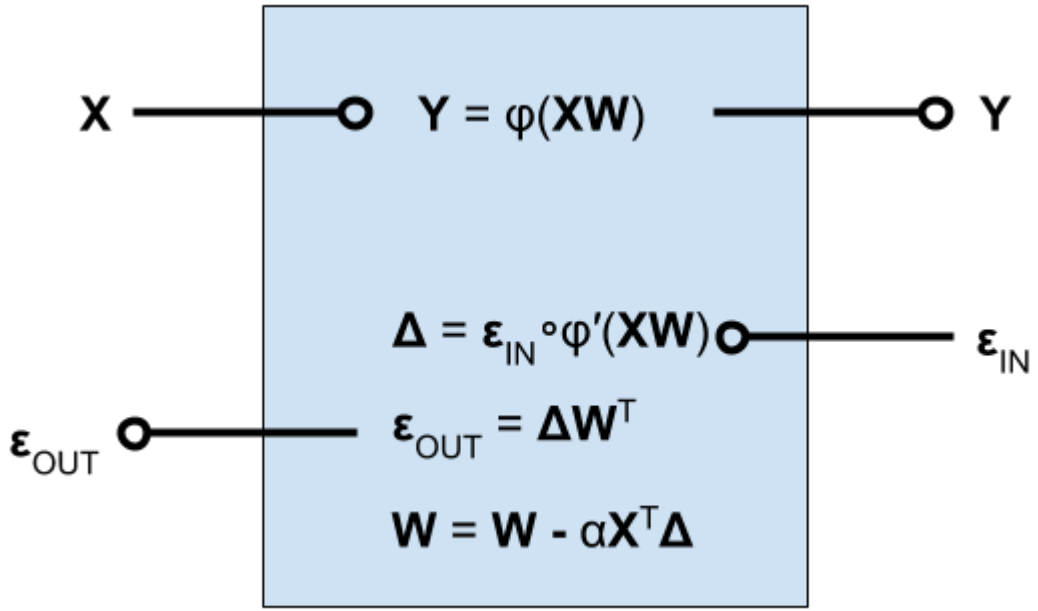


https://scikit-learn.org/stable/modules/neural_networks_supervised.html

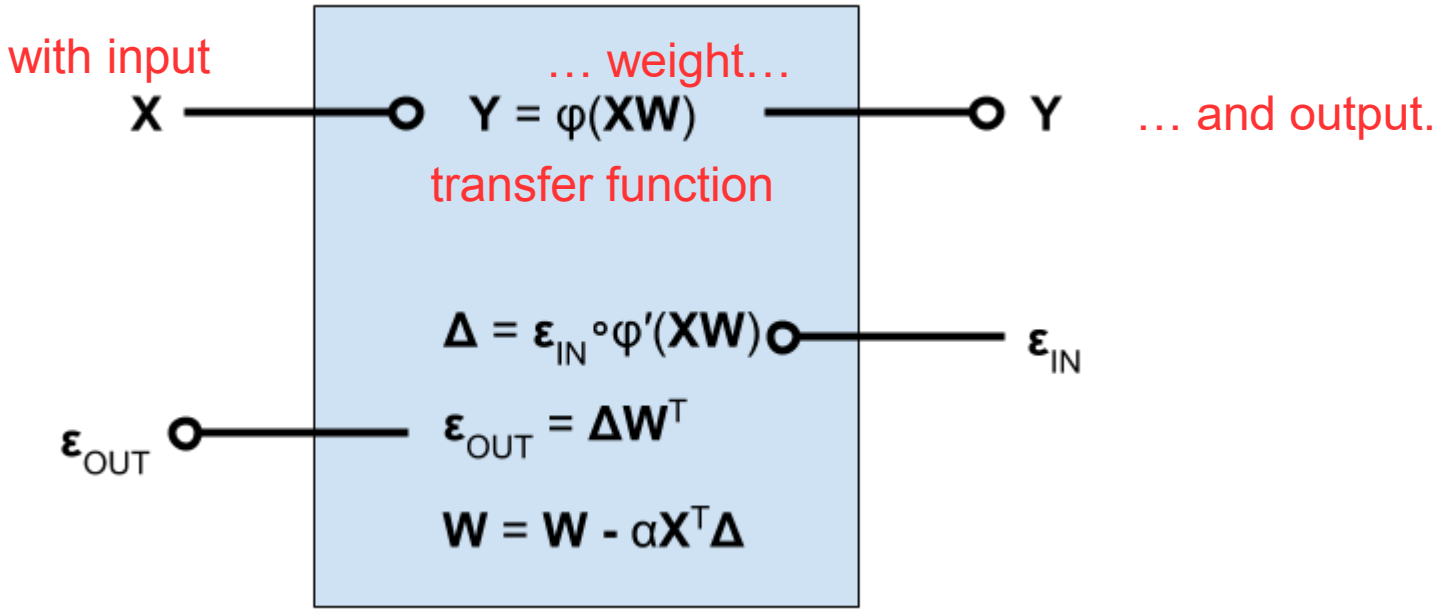
Instead of viewing an MLP with dots and sticks ...

let’s view it as a super-position of linear models

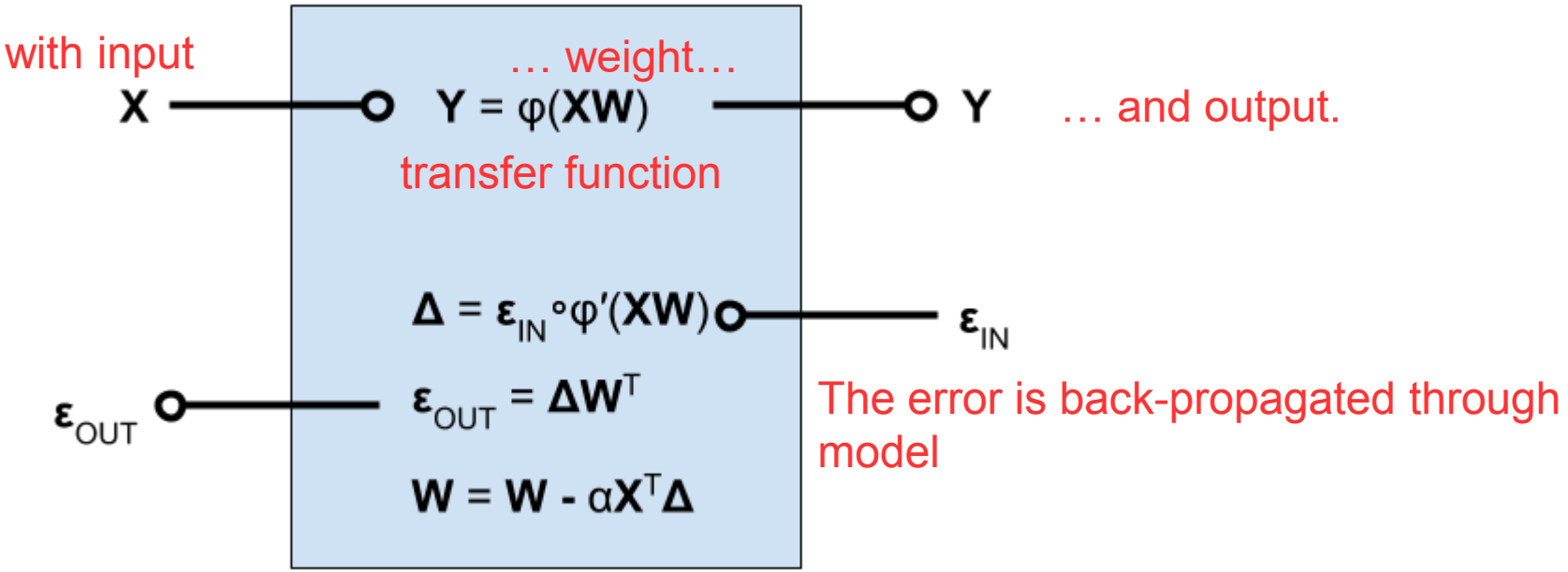
A “Neural Layer” is just one Linear Model



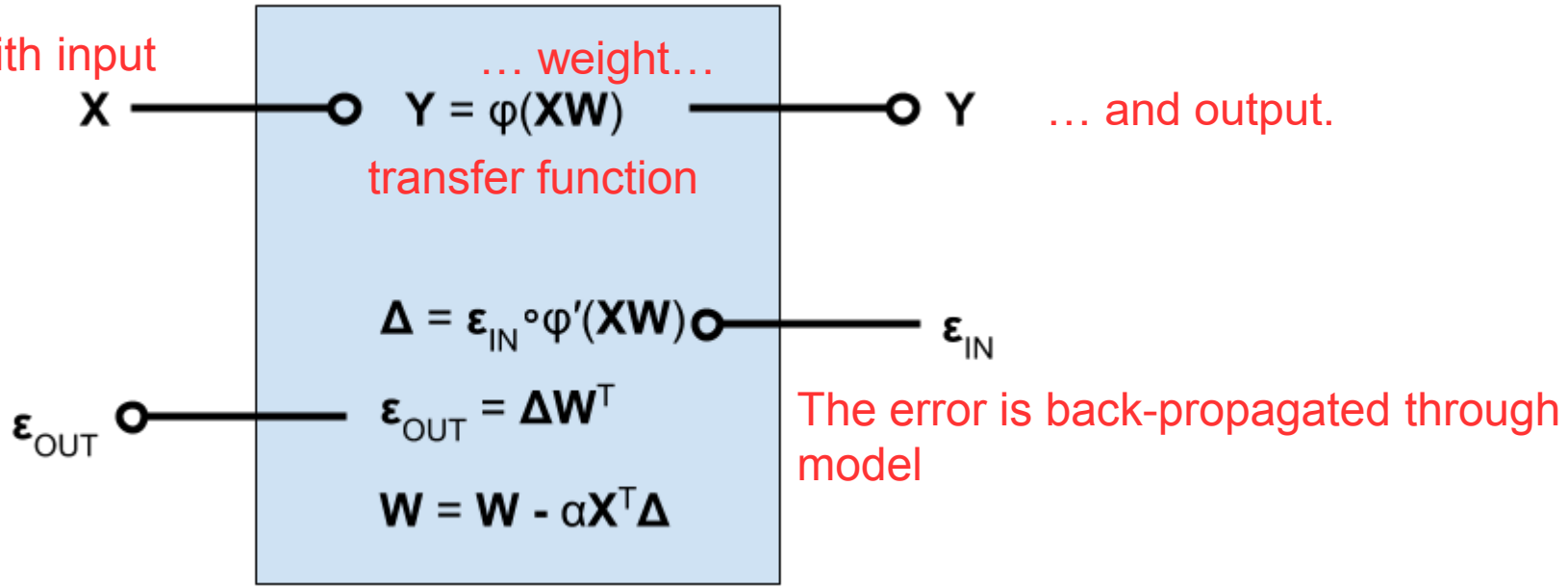
A “Neural Layer” is just one Linear Model …



A “Neural Layer” is just one Linear Model …



A “Neural Layer” is just one Linear Model …

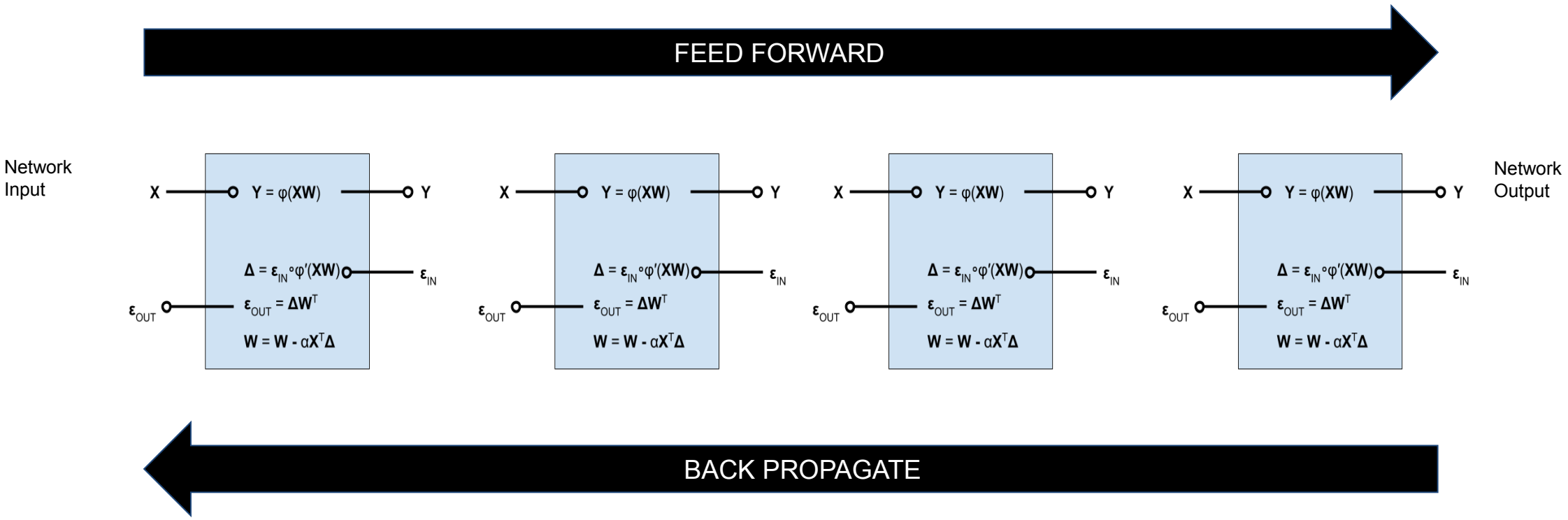


The parameters can be updated using any optimizer and form desired for that layer.

```
public class LinearModel {

    private RealMatrix weight;
    private RealVector bias;
    private final OutputFunction outputFunction;

    // Constructor goes here

    public RealMatrix getOutput(RealMatrix input) {

        return outputFunction.getOutput(input, weight, bias);

    }

    public void setWeight(RealMatrix weight) {
        this.weight = weight;
    }

    public void setBias(RealVector bias) {
        this.bias = bias;
    }

}
```

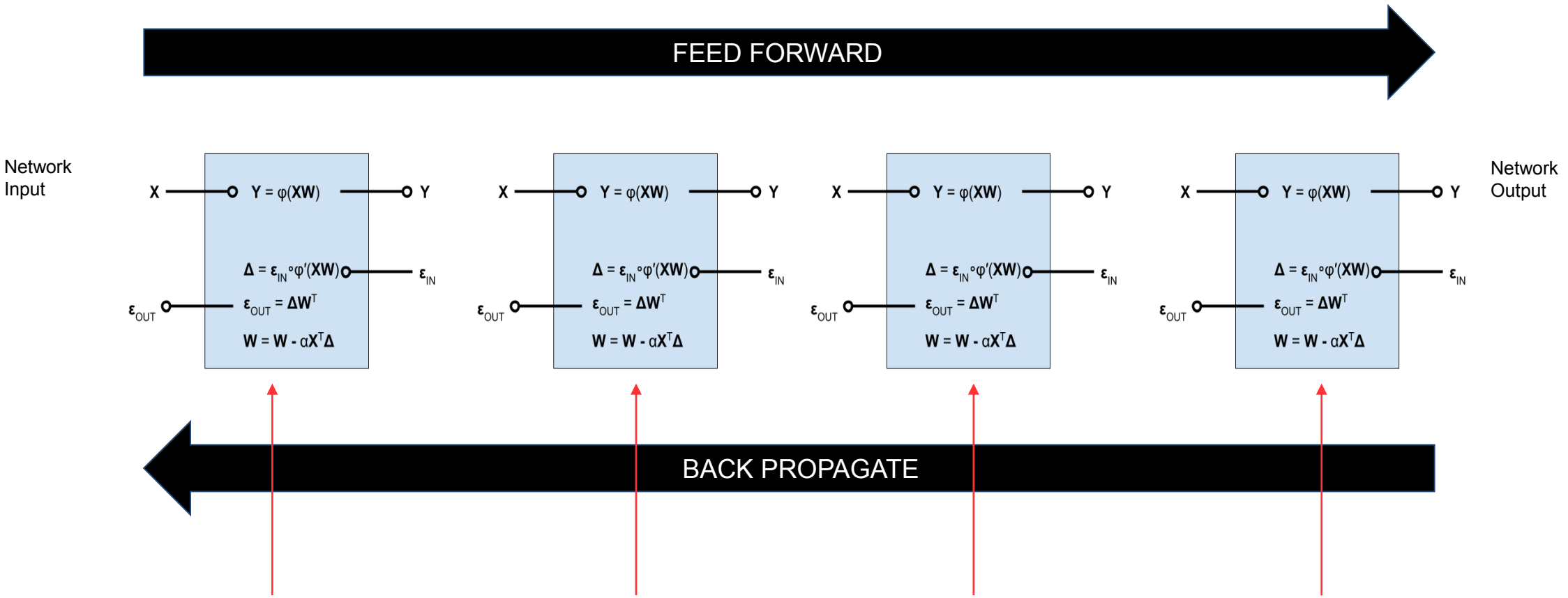
```
public class NetworkLayer extends LinearModel {

    RealMatrix input;
    RealMatrix inputError;
    RealMatrix output;
    RealMatrix outputError;
    Optimizer optimizer;

    // Constructor goes here

    public void update() {

            // back propagate error
            // and then update weights

    }

}
```

Code in book is at:
https://github.com/oreillymedia/Data_Science_with_Java


FEED FORWARD
Network Input
X
Y = φ(XW)
Y
Δ = ε_IN °φ′(XW)
ε_IN
ε_OUT
ε_OUT = ΔW^T
W = W - αX^TΔ
Network Output
BACK PROPAGATE

```
public class DeepNetwork extends IterativeLearningProcess {

    private List<NetworkLayer> layers;

   @Override
    public RealMatrix predict(RealMatrix input) {

        for (NetworkLayer layer : layers) {

            /* calc the output and set to next layer input*/

        }

        return output;

    }

    @Override
    protected void update(RealMatrix input, RealMatrix target, RealMatrix output) {

        /* create list iterator and set cursor to last! */
        ListIterator li = layers.listIterator(layers.size());

        while (li.hasPrevious()) {

            NetworkLayer layer = (NetworkLayer) li.previous();

           // back propagate error

        }
    }
}
```

Code in book is at:
https://github.com/oreillymedia/Data_Science_with_Java



Given the Java architecture described here, NOTE how easy it is to inspect each layer, and to define unique optimizer, update schema and others PER LAYER!

Iris Example:

4 input features and 3 output classes

Much easier to access / control layers with following example

Note different optimizer rules for different layers

Code in book is at: https://github.com/oreillymedia/Data_Science_with_Java

```
        Iris iris = new Iris();

        RealMatrix data = iris.getFeatures();
        RealMatrix target = iris.getLabels();
        MatrixResampler mr = new MatrixResampler(data, iris.getLabels());
        mr.calculateTestTrainSplit(0.40);

        DeepNetwork net = new DeepNetwork();
//        net.addLayer(new NetworkLayer(4, 7, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
//        net.addLayer(new NetworkLayer(7, 3, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
        net.addLayer(new NetworkLayer(4, 7, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(2, 2, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(10, 50, new TanhOutputFunction(), new GradientDescent(0.001)));
//        net.addLayer(new NetworkLayer(50, 5, new TanhOutputFunction(), new GradientDescent(0.001)));
        net.addLayer(new NetworkLayer(7, 3, new SoftmaxOutputFunction(), new GradientDescent(0.001)));
        net.setLossFunction(new SoftMaxCrossEntropyLossFunction());
//        net.setLossFunction(new QuadraticLossFunction());
        net.setBatchSize(1);
        net.setMaxIterations(600000);
        net.setPrecision(10E-9);

        net.learn(mr.getTrainingFeatures(), mr.getTrainingLabels());
        RealMatrix predictions = net.predict(mr.getTestingFeatures());
        ClassifierAccuracy acc = new ClassifierAccuracy(predictions, mr.getTestingLabels());

        System.out.println("converged = " + net.isConverged());
        System.out.println("iterations = " + net.getNumIterations());
        System.out.println("accuracy = " + acc.getAccuracy());
```

Code in book is at: https://github.com/oreillymedia/Data_Science_with_Java

```
        Iris iris = new Iris();

        RealMatrix data = iris.getFeatures();
        RealMatrix target = iris.getLabels();
        MatrixResampler mr = new MatrixResampler(data, iris.getLabels());
        mr.calculateTestTrainSplit(0.40);

        DeepNetwork net = new DeepNetwork();
//        net.addLayer(new NetworkLayer(4, 7, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
//        net.addLayer(new NetworkLayer(7, 3, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
        net.addLayer(new NetworkLayer(4, 7, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(2, 2, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(10, 50, new TanhOutputFunction(), new GradientDescent(0.001)));
//        net.addLayer(new NetworkLayer(50, 5, new TanhOutputFunction(), new GradientDescent(0.001)));
        net.addLayer(new NetworkLayer(7, 3, new SoftmaxOutputFunction(), new GradientDescent(0.001)));
        net.setLossFunction(new SoftMaxCrossEntropyLossFunction());
//        net.setLossFunction(new QuadraticLossFunction());
        net.setBatchSize(1);
        net.setMaxIterations(600000);
        net.setPrecision(10E-9);

        net.learn(mr.getTrainingFeatures(), mr.getTrainingLabels());
        RealMatrix predictions = net.predict(mr.getTestingFeatures());
        ClassifierAccuracy acc = new ClassifierAccuracy(predictions, mr.getTestingLabels());

        System.out.println("converged = " + net.isConverged());
        System.out.println("iterations = " + net.getNumIterations());
        System.out.println("accuracy = " + acc.getAccuracy());
```

Code in book is at: https://github.com/oreillymedia/Data_Science_with_Java

```
        Iris iris = new Iris();

        RealMatrix data = iris.getFeatures();
        RealMatrix target = iris.getLabels();
        MatrixResampler mr = new MatrixResampler(data, iris.getLabels());
        mr.calculateTestTrainSplit(0.40);

        DeepNetwork net = new DeepNetwork();
//        net.addLayer(new NetworkLayer(4, 7, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
//        net.addLayer(new NetworkLayer(7, 3, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
        net.addLayer(new NetworkLayer(4, 7, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(2, 2, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(10, 50, new TanhOutputFunction(), new GradientDescent(0.001)));
//        net.addLayer(new NetworkLayer(50, 5, new TanhOutputFunction(), new GradientDescent(0.001)));
        net.addLayer(new NetworkLayer(7, 3, new SoftmaxOutputFunction(), new GradientDescent(0.001)));
        net.setLossFunction(new SoftMaxCrossEntropyLossFunction());
//        net.setLossFunction(new QuadraticLossFunction());
        net.setBatchSize(1);
        net.setMaxIterations(600000);
        net.setPrecision(10E-9);

        net.learn(mr.getTrainingFeatures(), mr.getTrainingLabels());
        RealMatrix predictions = net.predict(mr.getTestingFeatures());
        ClassifierAccuracy acc = new ClassifierAccuracy(predictions, mr.getTestingLabels());

        System.out.println("converged = " + net.isConverged());
        System.out.println("iterations = " + net.getNumIterations());
        System.out.println("accuracy = " + acc.getAccuracy());
```

Code in book is at: https://github.com/oreillymedia/Data_Science_with_Java

```
        Iris iris = new Iris();

        RealMatrix data = iris.getFeatures();
        RealMatrix target = iris.getLabels();
        MatrixResampler mr = new MatrixResampler(data, iris.getLabels());
        mr.calculateTestTrainSplit(0.40);

        DeepNetwork net = new DeepNetwork();
//        net.addLayer(new NetworkLayer(4, 7, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
//        net.addLayer(new NetworkLayer(7, 3, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
        net.addLayer(new NetworkLayer(4, 7, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(2, 2, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(10, 50, new TanhOutputFunction(), new GradientDescent(0.001)));
//        net.addLayer(new NetworkLayer(50, 5, new TanhOutputFunction(), new GradientDescent(0.001)));
        net.addLayer(new NetworkLayer(7, 3, new SoftmaxOutputFunction(), new GradientDescent(0.001)));
        net.setLossFunction(new SoftMaxCrossEntropyLossFunction());
//        net.setLossFunction(new QuadraticLossFunction());
        net.setBatchSize(1);
        net.setMaxIterations(600000);
        net.setPrecision(10E-9);

        net.learn(mr.getTrainingFeatures(), mr.getTrainingLabels());
        RealMatrix predictions = net.predict(mr.getTestingFeatures());
        ClassifierAccuracy acc = new ClassifierAccuracy(predictions, mr.getTestingLabels());

        System.out.println("converged = " + net.isConverged());
        System.out.println("iterations = " + net.getNumIterations());
        System.out.println("accuracy = " + acc.getAccuracy());
```

Code in book is at: https://github.com/oreillymedia/Data_Science_with_Java

```
        Iris iris = new Iris();

        RealMatrix data = iris.getFeatures();
        RealMatrix target = iris.getLabels();
        MatrixResampler mr = new MatrixResampler(data, iris.getLabels());
        mr.calculateTestTrainSplit(0.40);

        DeepNetwork net = new DeepNetwork();
//        net.addLayer(new NetworkLayer(4, 7, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
//        net.addLayer(new NetworkLayer(7, 3, new RampOutputFunction(), new GradientDescentMomentum(.001, 0.95)));
        net.addLayer(new NetworkLayer(4, 7, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(2, 2, new TanhOutputFunction(), new GradientDescent(.001)));
//        net.addLayer(new NetworkLayer(10, 50, new TanhOutputFunction(), new GradientDescent(0.001)));
//        net.addLayer(new NetworkLayer(50, 5, new TanhOutputFunction(), new GradientDescent(0.001)));
        net.addLayer(new NetworkLayer(7, 3, new SoftmaxOutputFunction(), new GradientDescent(0.001)));
        net.setLossFunction(new SoftMaxCrossEntropyLossFunction());
//        net.setLossFunction(new QuadraticLossFunction());
        net.setBatchSize(1);
        net.setMaxIterations(600000);
        net.setPrecision(10E-9);

        net.learn(mr.getTrainingFeatures(), mr.getTrainingLabels());
        RealMatrix predictions = net.predict(mr.getTestingFeatures());
        ClassifierAccuracy acc = new ClassifierAccuracy(predictions, mr.getTestingLabels());

        System.out.println("converged = " + net.isConverged());
        System.out.println("iterations = " + net.getNumIterations());
        System.out.println("accuracy = " + acc.getAccuracy());
```

- That didn’t seem too bad, did it???

Summary: Data Science with Java?

Java is fairly universal world-wide and there is no shortage of programming talent.

Java powers trust-worthy infrastructure applications already.

Perhaps Java coding practices / culture leads to better designed and documented code?

Breaking pieces into logical mathematical units (e.g. neural layers example) is natural in OOP. Easier to inspect, update and understand code. For example, the Java designation of “one class per file.”

Well defined input, black-box and output types is a GREAT IDEA. There should never be surprises!

Java docs are easy to read and understand partly because OOP encourages cleaner design.

JVM was designed with security in mind, since 1995.

```
System.exit(0);
```