rfig 1.1 tutorial (incomplete)

programming figures/slides in Ruby

Percy Liang

Title of the slide goes here

We can start writing text...

Each string starts a new line.

Let's make this centered

Still centered

Back to left justified

Right justified

All text is treated as LaTeX, so math is easy to do: $\frac{1}{2} - \pi$

By default, text does not autowrap. If it is put into an autowrap function we can make it wrap up to a certain width (10 inches) and obey a certain orientation (flushfull).

Animation

Each slide can produce many PDF pages.

First I appear

Animation

Each slide can produce many PDF pages.

First I appear

I appear on the next slide

Animation

Each slide can produce many PDF pages.

First I appear

I appear on the next slide

and so on...

Animation 2

Another way to animate is to specify the levels at which objects appear:

level 0

Animation 2

Another way to animate is to specify the levels at which objects appear:

level 0

level 1

Animation 2

Another way to animate is to specify the levels at which objects appear:

level 2

level 0

level 1

Animation 2

Another way to animate is to specify the levels at which objects appear:

level 2

level 3

level 0

level 1

Transformations

Normal size

Smaller

Rotated

Slanted

The color can be changed

Many transformations can be strung together

Lists

- Itemized lists are easy to make
- We can also create hierarchical lists:
  - Sub bullet 1
  - Sub bullet 2

1. Now we can number the points
2. See the numbers increase

Tables

So far, we have just dumped content in a sequential manner.

We would like to format our slides somehow.

There are two ways of doing that: using tables and overlays.

Here is a basic table:

first row, first column	second column
second row now	last one

We can justify the table and remove the border:

aaa	d	e
b	ccc	fff

If we want tables with just one row or one column,

we can use the following shorthand:

row 1
row 2

column 1	column 2

Overlays

Using overlays, we can place things on top of each other.

The pivot specifies the relative positions

that should be used to align the objects in the overlay.

~~$0 = 1$~~

the elements

Overlays

Using overlays, we can place things on top of each other.
The pivot specifies the relative positions
that should be used to align the objects in the overlay.
~~$0 = 1$~~

in this

Overlays

Using overlays, we can place things on top of each other.

The pivot specifies the relative positions

that should be used to align the objects in the overlay.

~~$0 = 1$~~

overlay should be centered

Overlays

Using overlays, we can place things on top of each other.

The pivot specifies the relative positions

that should be used to align the objects in the overlay.

~~$0 = 1$~~

overlay should be centered

whereas the ones

Overlays

Using overlays, we can place things on top of each other.

The pivot specifies the relative positions

that should be used to align the objects in the overlay.

~~$0 = 1$~~

overlay should be centered

here

Overlays

Using overlays, we can place things on top of each other.

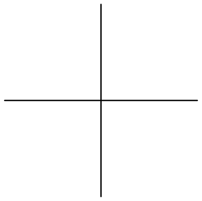
The pivot specifies the relative positions

that should be used to align the objects in the overlay.

~~$0 = 1$~~

overlay should be centered

should be right justified

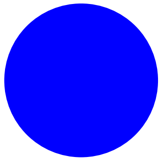


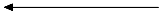
Formatting the slide

We can also change properties of the slide.
Compare this slide
with the others to see what has changed.

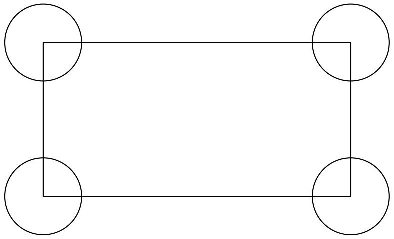
Graphics

We can integrate figures into our slides fairly easily.

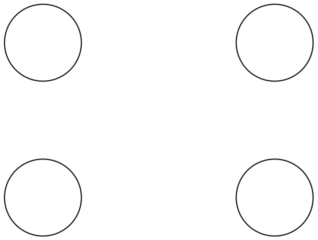
Here is a circle: 

We can draw an arrow: 

We can go low-level and specify absolute positions:

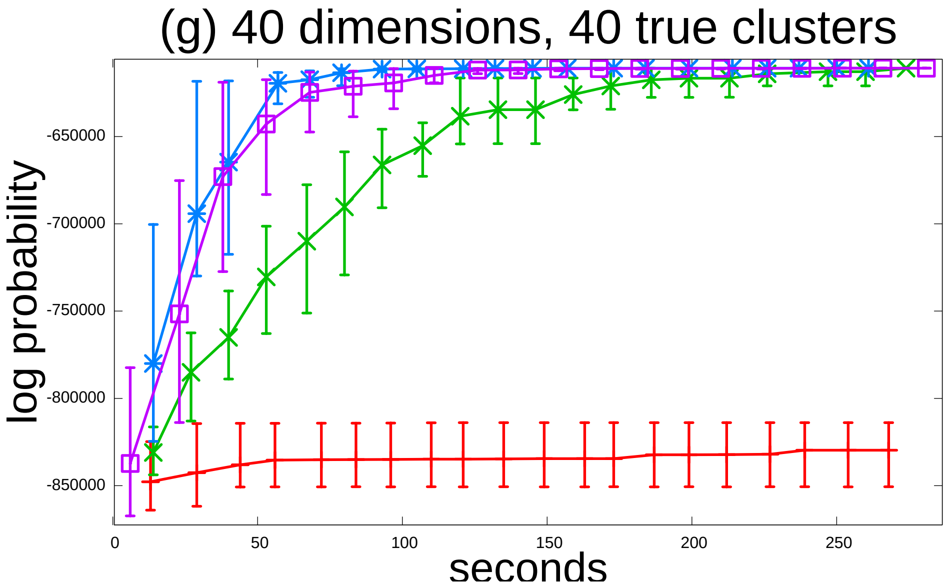


However, it's much cleaner and modular to use tables when possible:



Imported graphics

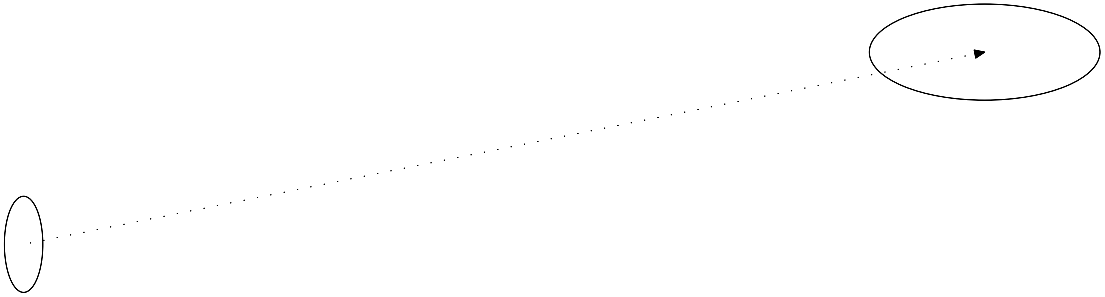
One can import jpeg and pdf files:



Flexible relative positioning of objects

We can specify the positions of some objects with respect to others.

A useful primitive to have is to connect two objects with a line:



We can also circle objects:

box ellipse

Tables example

A more complicated example of using lots of features:

1 2 3 4		
1	3	4

1	2
3	4

Tables example

A more complicated example of using lots of features:

1 2	3 4
1 2 3 4	
1	2
3	4

Overlays

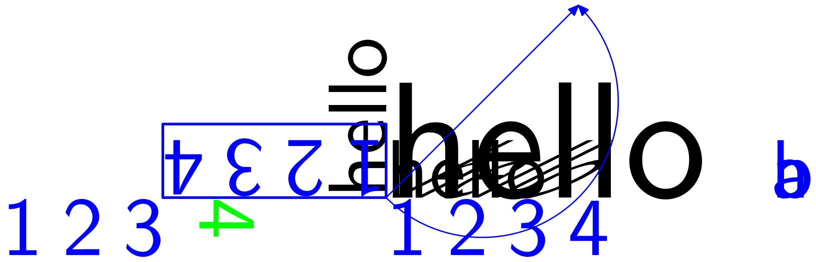
hello

hello

hello

hello

Overlays



1 2 3 4

1 2 3 4

hello

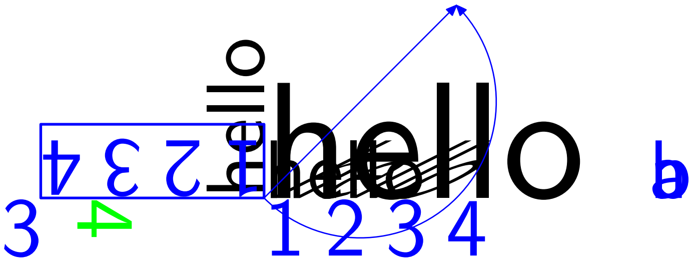
hello

hello

1 2 3 4

a b

Overlays



1 2 3 4

1 2 3 4

hello

hello

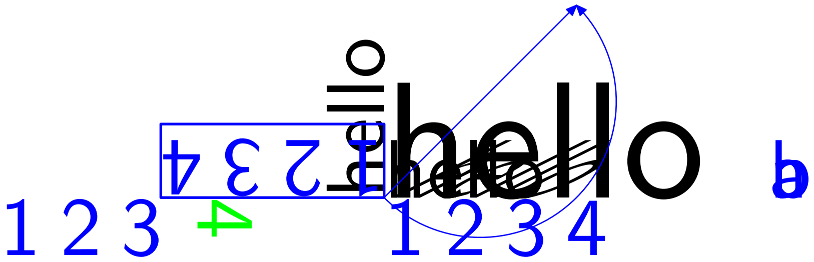
hello

1 2 3 4

b a

hello

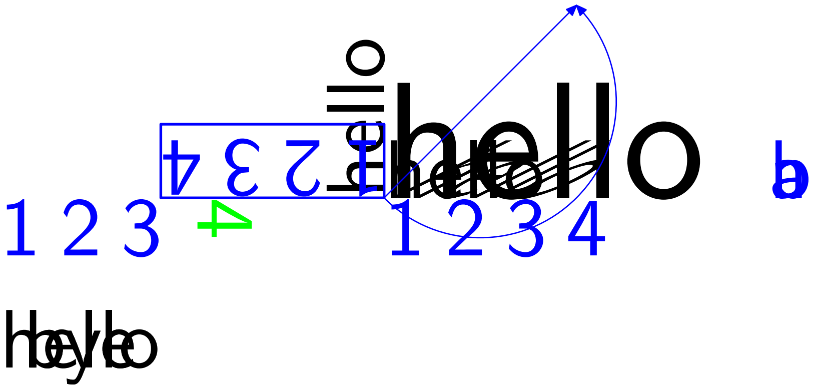
Overlays



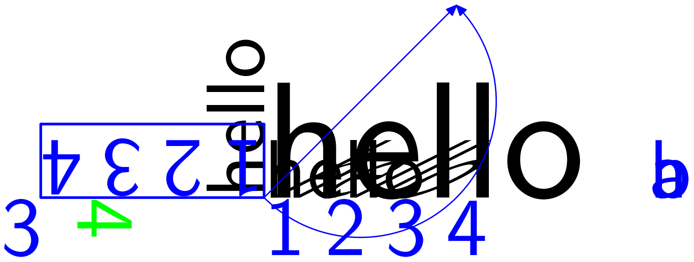
hello

bye

Overlays



Overlays



1 2 3 4

1 2 3 4

hello

hello

hello

1 2 3 4

a b

hello bye

done

Encircling things

redEncircle new object

existing object

Encircling things

redEncircle new object

existing object

existing object

encircle this new object

enrect this new object

Lists

1. one

Lists

1. one
3. two
4. two dot one

Lists

1. one

3. two

4. two dot one

c. two dot two

5. three

Lists

1. one
3. two
4. two dot one
   c. two dot two
5. three
7. four

Lists

hello

1. one

3. two

4. two dot one

c. two dot two

5. three

7. four

Drawing pictures and post-processing

here

Drawing pictures and post-processing

here

eh

shown early

Drawing pictures and post-processing

here

foo eh

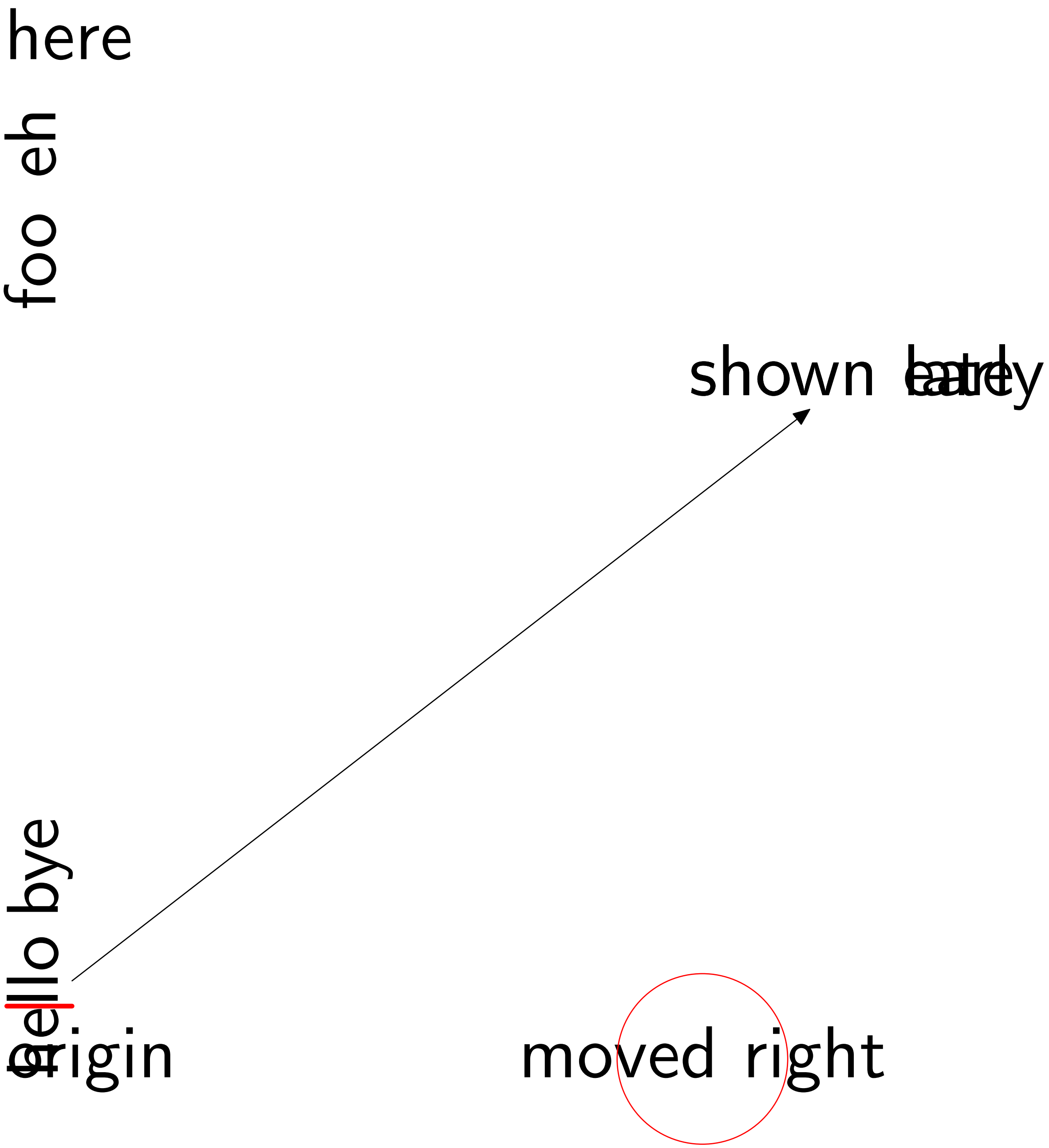
shown early

hello bye

origin

moved right

Drawing pictures and post-processing



here

foo eh

shown early

later

hello bye

origin

moved right

Pause levels and variables

A plain table:

1

Pause levels and variables

A plain table:

1	2	3
4	5	6

Pause levels and variables

A plain table:

1	2	3
4	5	6

A table with a border:

1	2	3
4	5	6

This is shown when the bordered table is

Pause levels and variables

A plain table:

1	2	3
4	5	6

A table with a border:

1	2	3
4	5	6

An overlay:

one

This is shown when the bordered table is

Pause levels and variables

A plain table:

1	2	3
4	5	6

A table with a border:

1	2	3
4	5	6

An overlay:

two

This is shown when the bordered table is

Pause levels and variables

A plain table:

1	2	3
4	5	6

A table with a border:

1	2	3
4	5	6

An overlay:

three

This is shown when the bordered table is

Pause levels and variables

A plain table:

1	2	3
4	5	6

A table with a border:

1	2	3
4	5	6

An overlay:

three

This is shown when the bordered table is

This is shown afterwards

More pause levels

Press any key to begin...

More pause levels

Press any key to begin...

We will show one column at a time

a

a

a

More pause levels

Press any key to begin...

We will show one column at a time

a b

a b

a b

More pause levels

Press any key to begin...

We will show one column at a time

a b c
a b c
a b c

More pause levels

Press any key to begin...

We will show one column at a time

a b c d
a b c d
a b c d

More pause levels

Press any key to begin...

We will show one column at a time

a	b	c	d
a	b	c	d
a	b	c	d

At the end

More pause levels

begin...

More pause levels

begin...
a

reset level

More pause levels

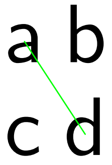
begin...
a b
c d
after in overlay
reset level reset level 2

More pause levels

begin...
a b
c d
after in overlay
reset level reset level 2

after out of overlay: printed after three pauses total

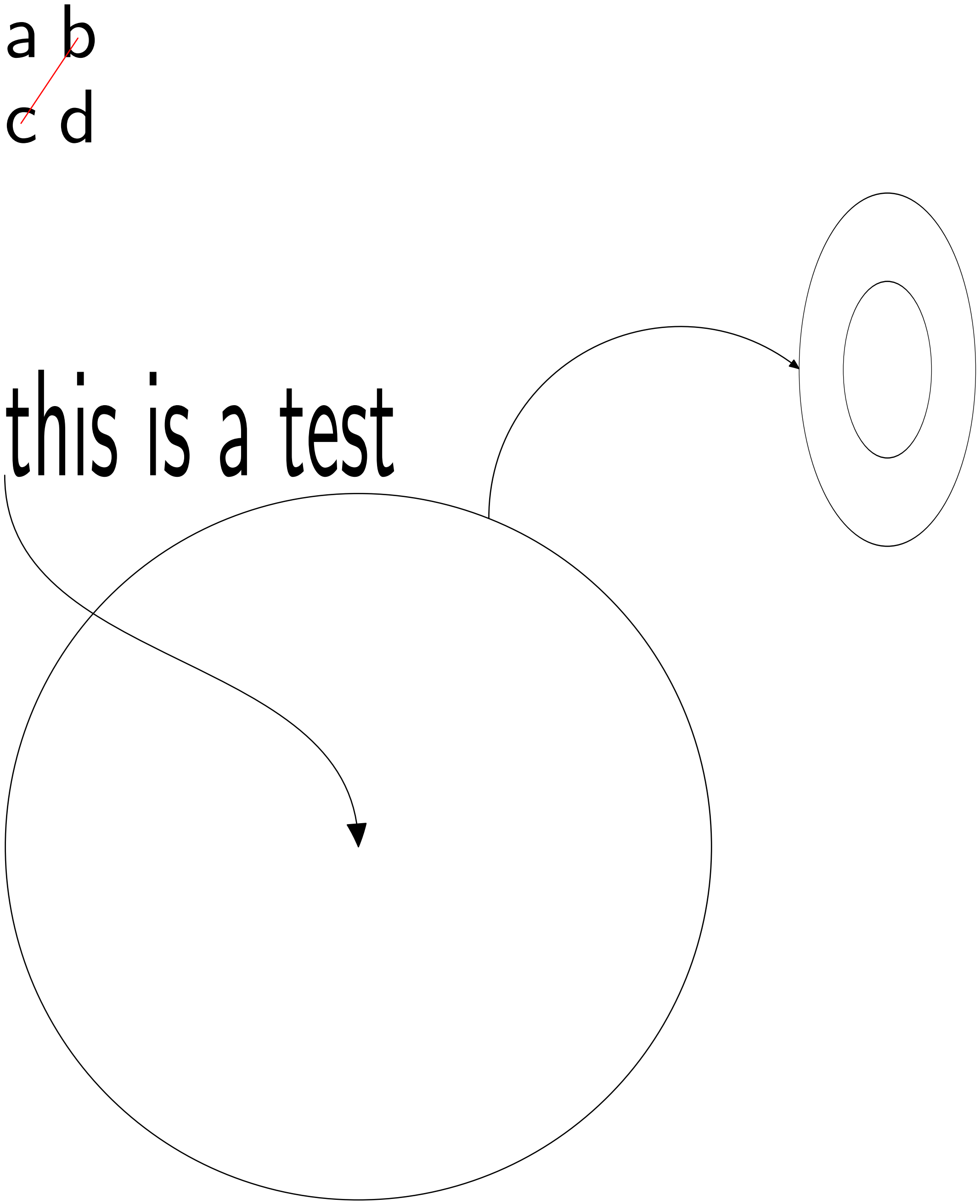
Referencing positions objects without postAdd



a b

c d

Referencing positions objects without postAdd



a b

c d

this is a test

Level strings

ab   e

Level strings

ab de

Level strings

abcde

Level strings

abcde

A next example in math:

Level strings

abcde

A next example in math:

$\underline{x+y}$

Level strings

abcde

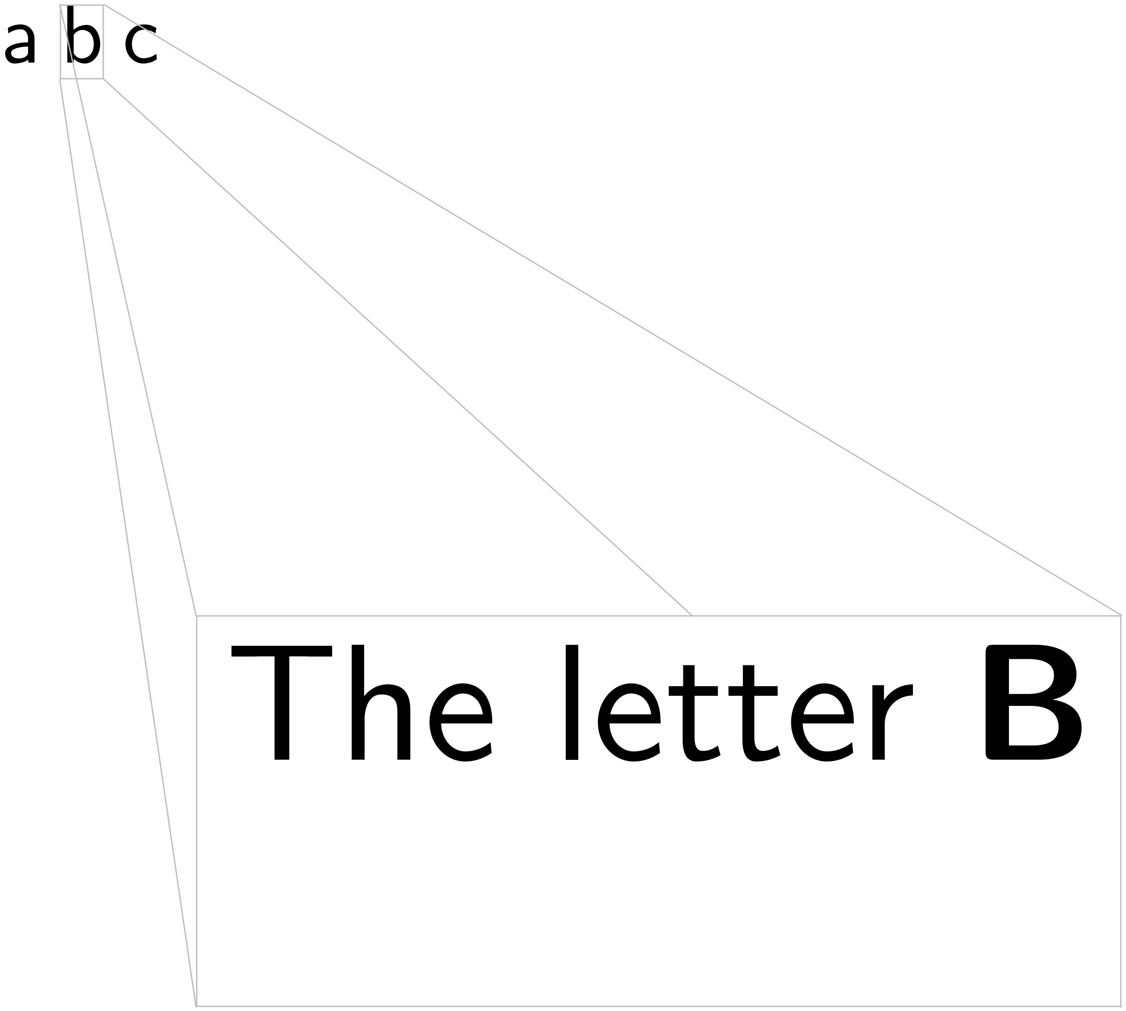
A next example in math:

$\frac{x+y}{z+w}$

Zooming

a b c

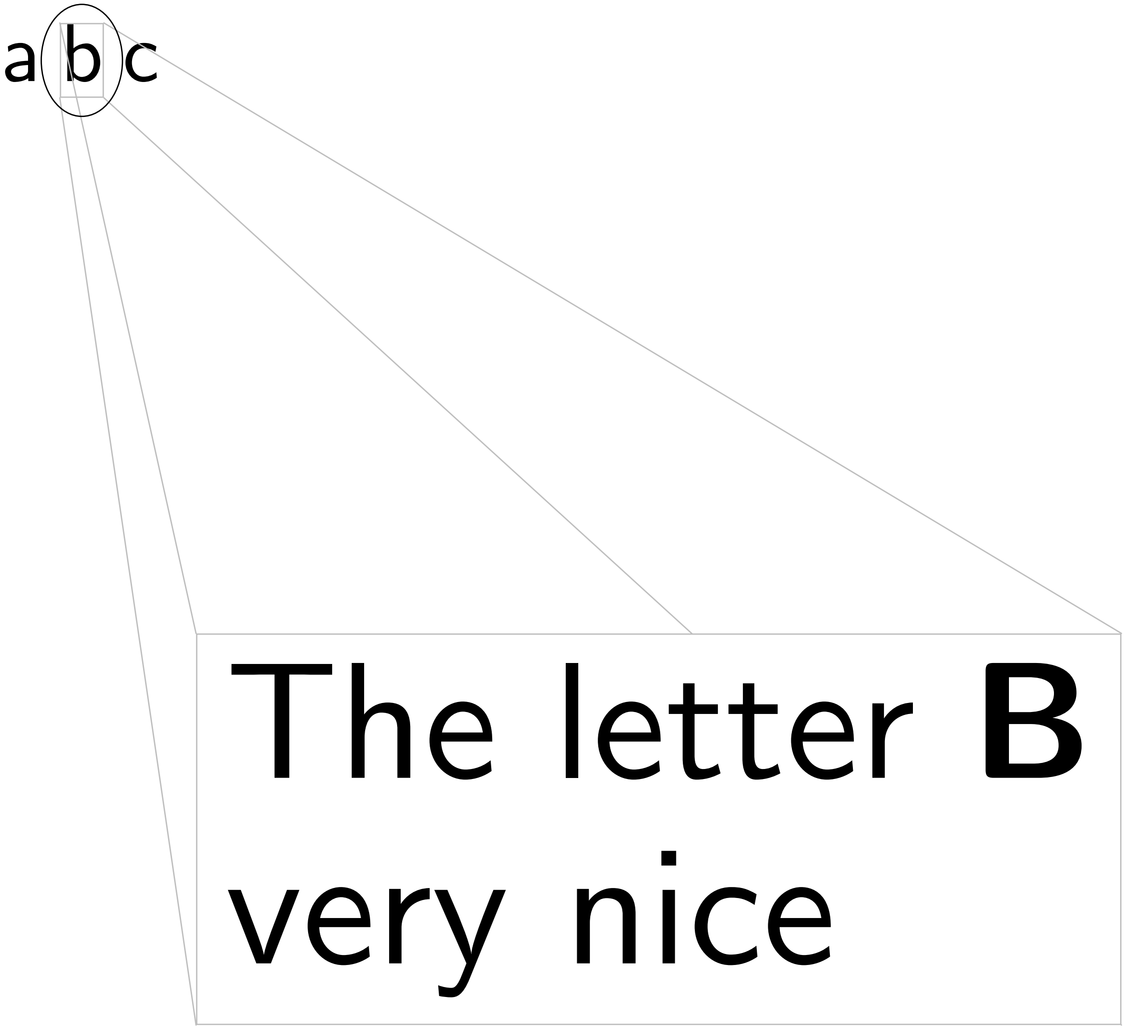
Zooming



a b c

The letter **B**

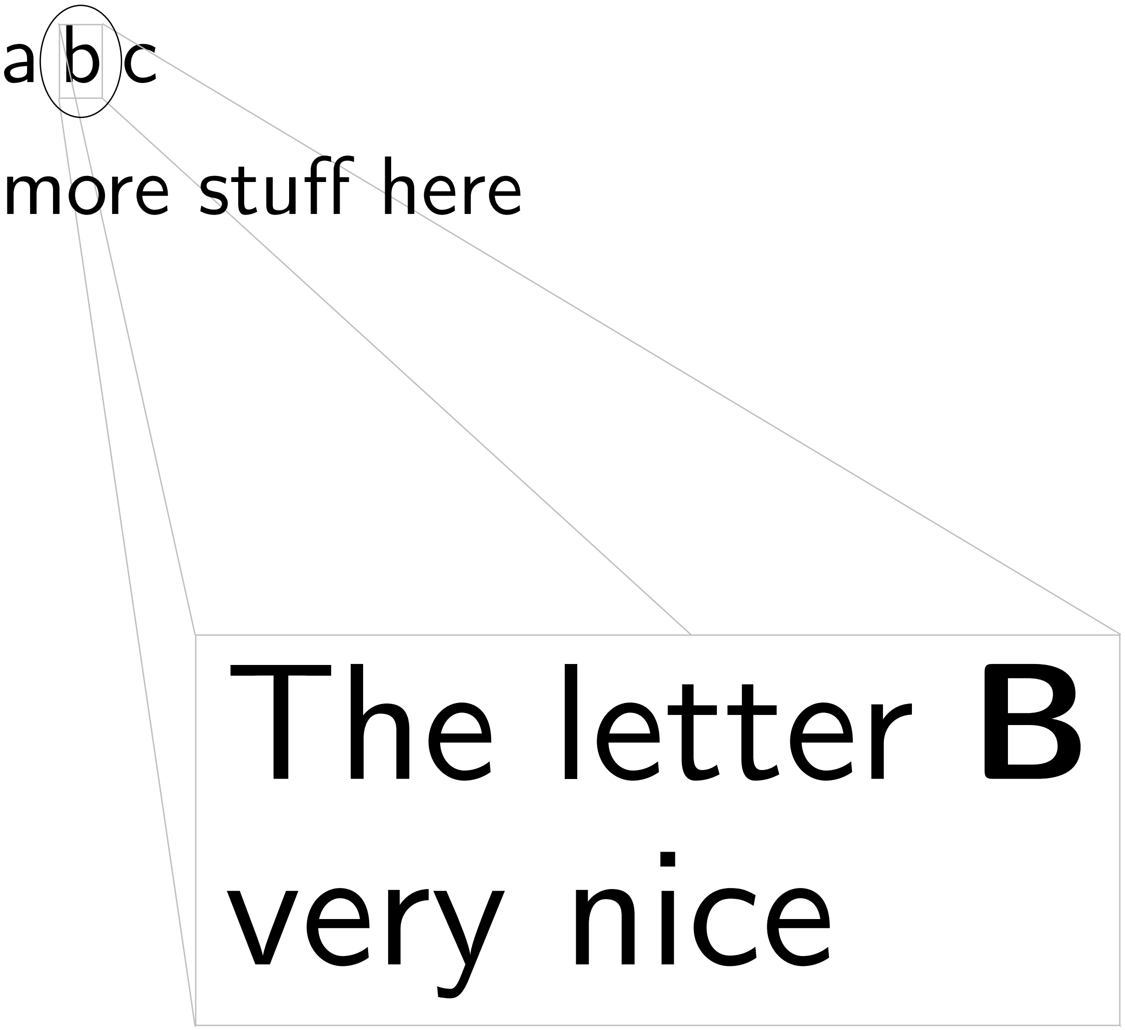
Zooming



a b c

The letter **B** very nice

Zooming



a b c

more stuff here

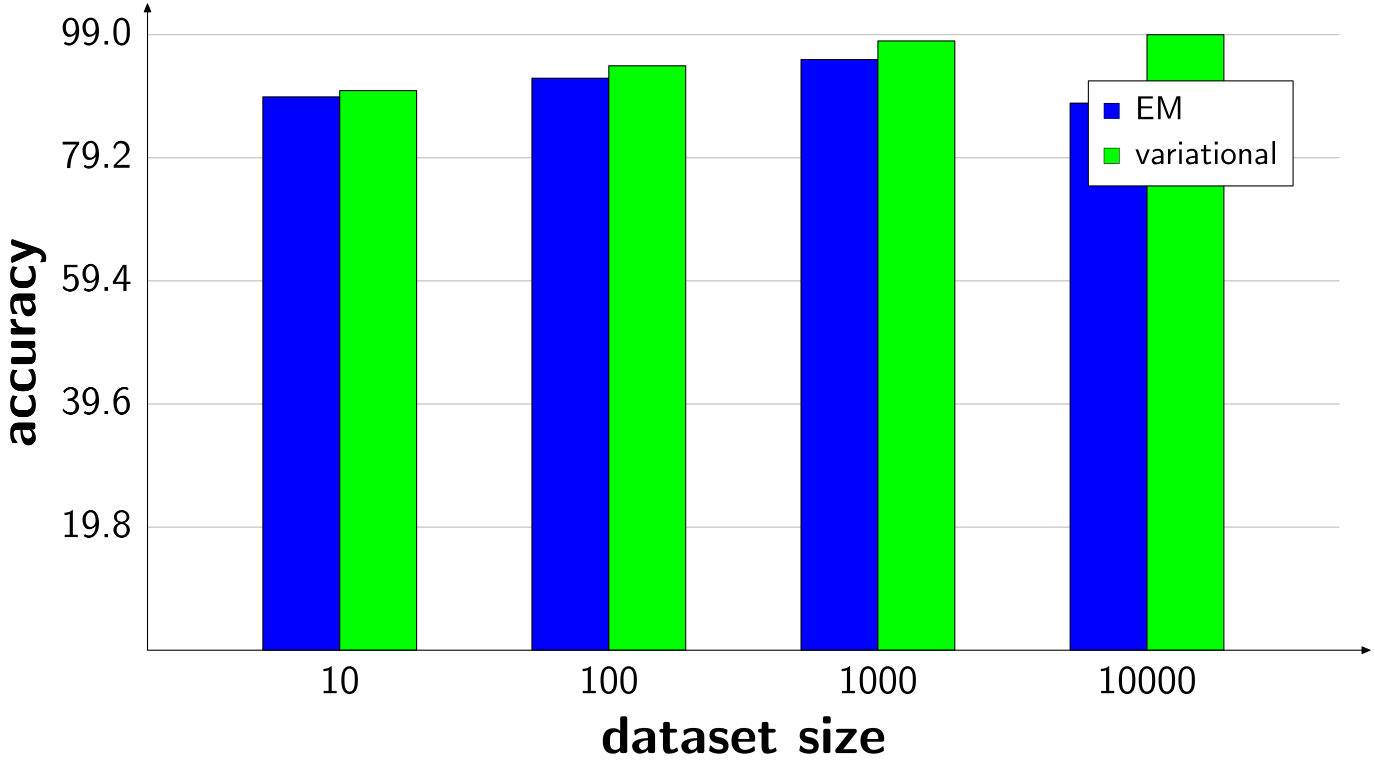
The letter **B**
very nice

Tables and graphs

method	10	100	1000	10000
EM	89	92	95	88
variational	90	94	98	99

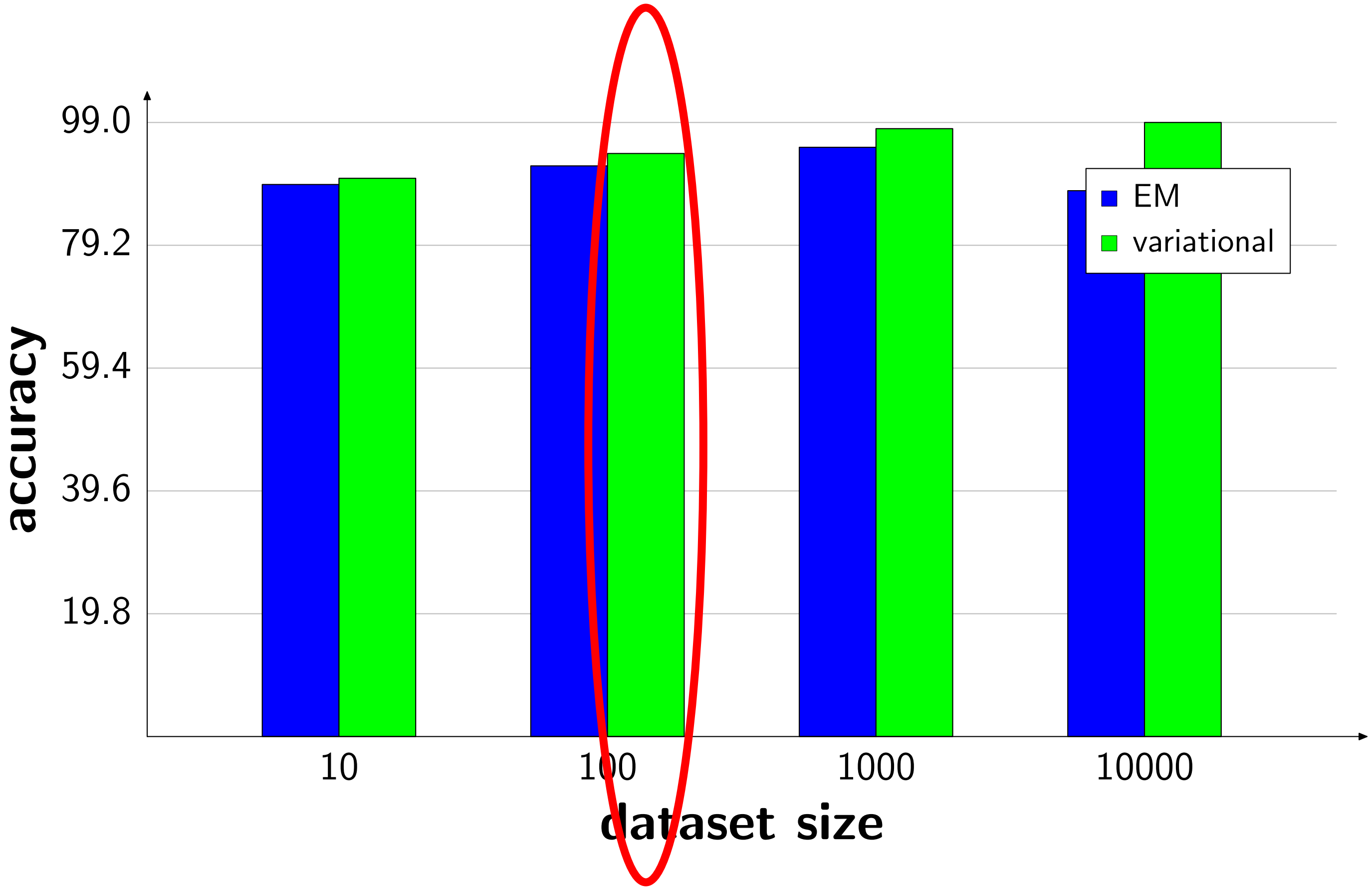
Tables and graphs

method	10	100	1000	10000
EM	89	92	95	88
variational	90	94	98	99



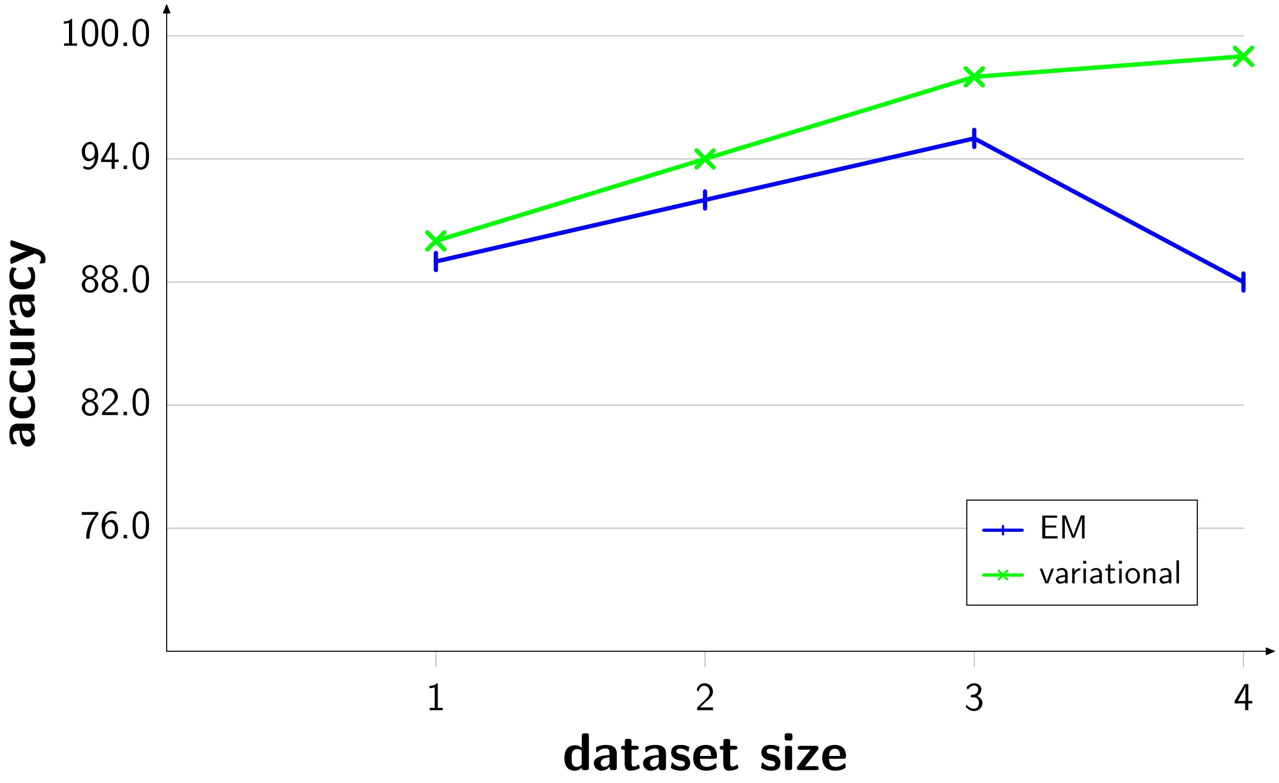
Tables and graphs

method	10	100	1000	10000
EM	89	92	95	88
variational	90	94	98	99



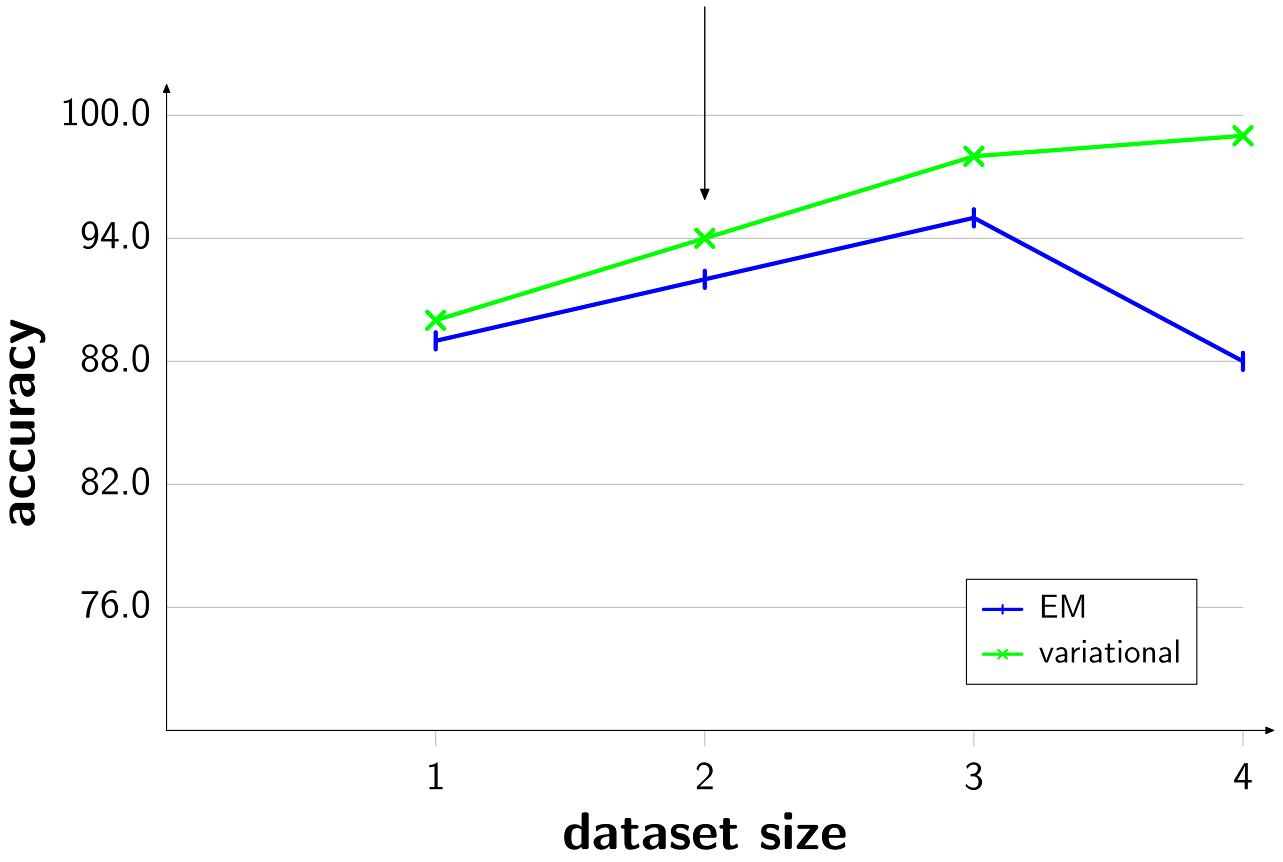
Tables and graphs

method	10	100	1000	10000
EM	89	92	95	88
variational	90	94	98	99



Tables and graphs

method	10	100	1000	10000
EM	89	92	95	88
variational	90	94	98	99



Interesting

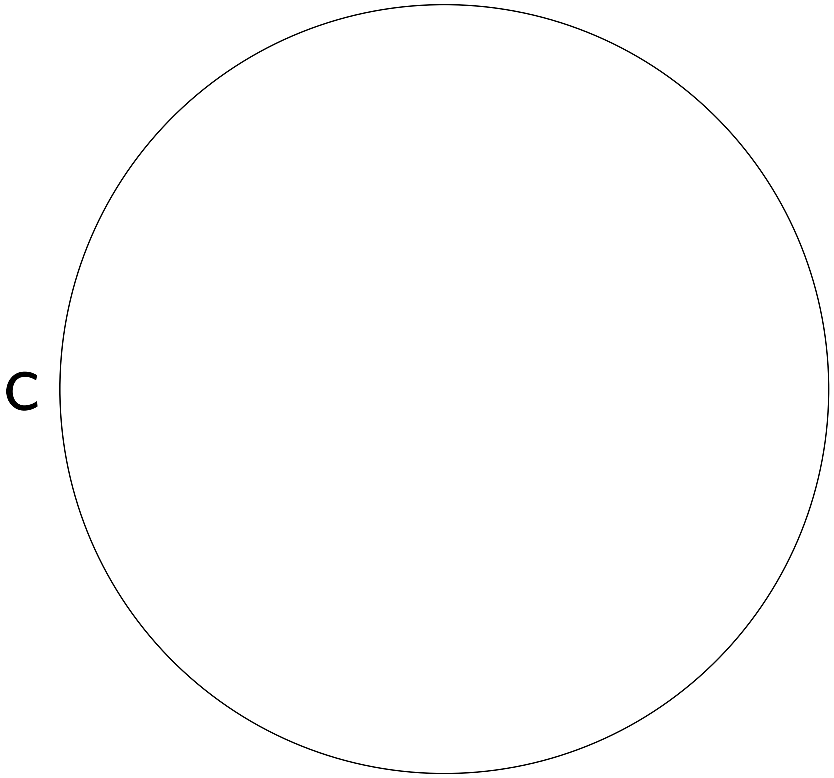
Changing bounds (on text)

Normal formatting:	Using raw bounds:	Using standard height:
begin	begin	begin
b	b	b
middle	middle	middle
d	d	d
end	end	end
d		

6 0.5 Raw bounds put g higher because it does not have the ascender. Using the standard height makes everything take as much vertical space as an a.

a

b


C

this should autowrap at 4 inches

if scaled by 0.5, autowrapping happens at 2

a

b

C

c

d