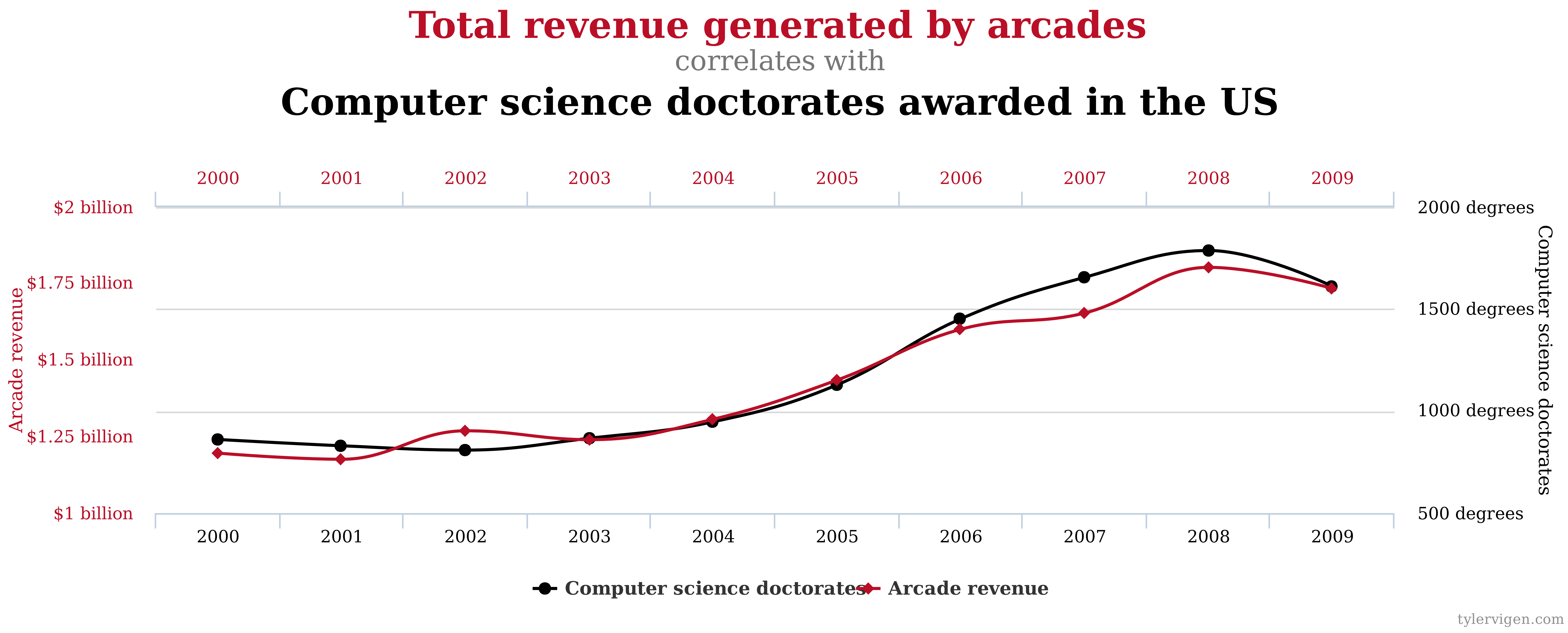
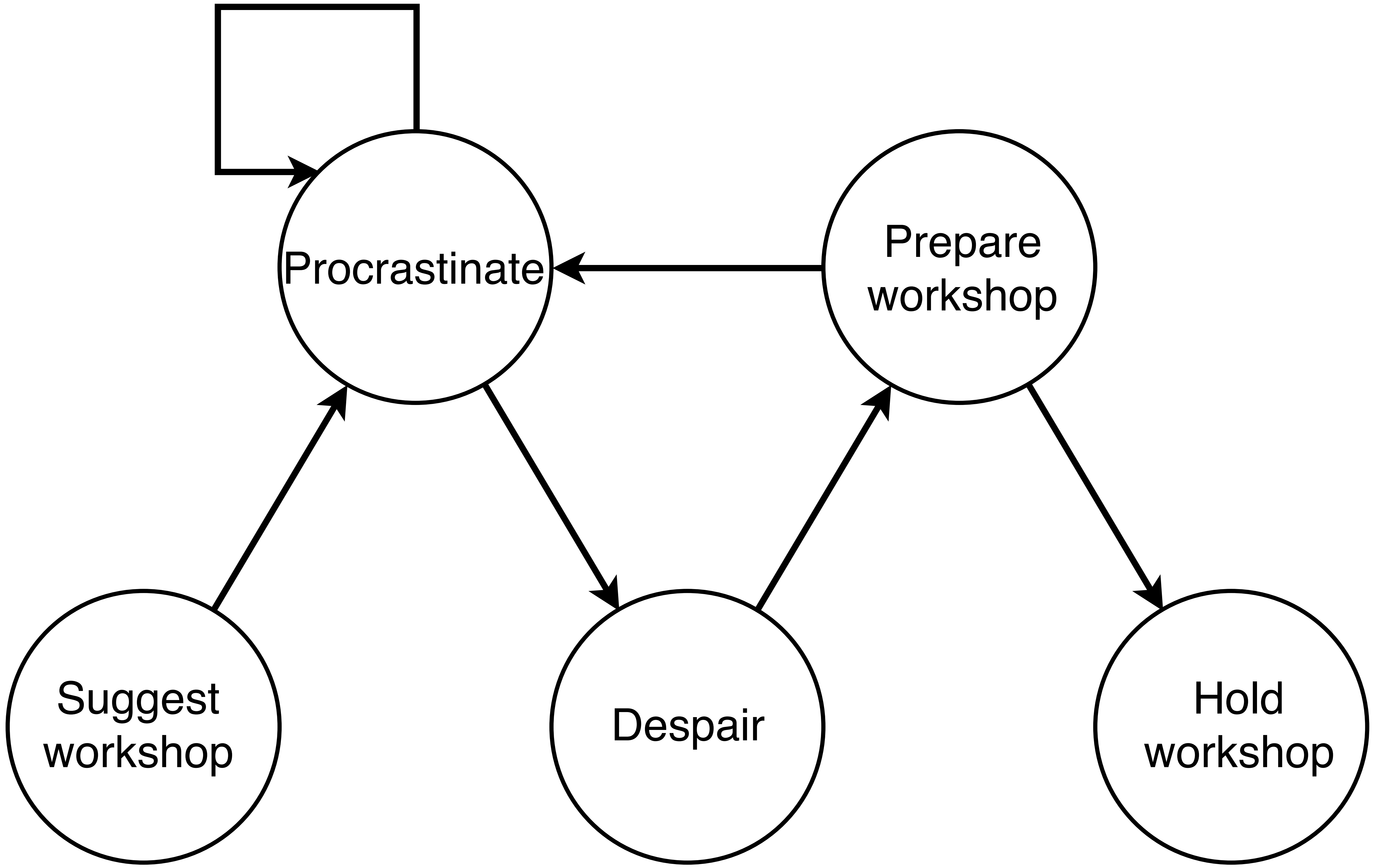
Tensorflow

Graph computation



Graph computation



Concepts

- Graph
- Session
- Tensor
- Constant, variable, placeholder
- Operation (op)

Graph

- Context for the content you define
- Rarely need for more than one

```
g = tf.Graph()
with g.as_default():
  # define content
```

Session

- Used for running a graph
- May evaluate whole or parts of graphs
- CPU or GPU
- Finds and evaluates dependencies in the graph automatically

```
sess = tf.Session()
output = sess.run(<something_from_graph>)
```

Tensor

- N-dimensional array
- Input, output and state is stored in tensors
- May be used as is, or through variables and constants

Constant

- Contains a tensor of constant value

```
data = [1, 2, 3]
constant = tf.constant(data)
out = sess.run(constant)
# out == [1, 2, 3]
```

Variable

- Mutable tensors that live in the graph
- Must be initialized

```
const_init = tf.constant_initializer(7)
var = tf.get_variable(
  name=‘my_variable’,
  shape=[3],
  initializer=const_init)
sess.run(tf.initialize_all_variables())
three_sevens = sess.run(var)
```

Placeholder

- Mechanism for providing input to the graph at runtime
- Works as a pointer to data outside the graph

```
input = tf.placeholder(
 dtype=tf.int32,
 shape=[3])
feed_dict = { input: [1, 2, 3] }
out = sess.run(input, feed_dict=feed_dict)
# the graph is evaluated with input == [1, 2, 3]
# out will here also evaluate to [1, 2, 3]
```

Operation

- Represents a piece of work (like a function)

- May have return values

```
operation = tf.initialize_all_variables()
sess.run(operation)
```

```
c = tf.constant([1, 2, 3])
# operation for summing c
sum = tf.reduce_sum(c)
out = sess.run(sum)
# out == 6
```

Activation

```
tf.tanh # Hyperbolic tangent, squashes (-1,1)
```

```
tf.sigmoid # Sigmoid, squashes (0,1)
```

```
tf.nn.softmax # Softmax ensures activations
  sum to 1.0.
  Typically applied for classification.
  Often (falsely) regarded as probabilities.
```

```
tf.relu # Rectified linear unit,
  lower bounds to 0.
```

Optimizer

```
tf.train.GradientDescentOptimizer
# implemented using backpropagation.
```

```
tf.train.RMSPropOptimizer
# good for deep networks
```