

Control logic

Hamish Gibbs

Tutorial #1: Conditional logic

- Conditional execution
- Core concepts:
 - Booleans (`True`, `False`)

```
1 x = 5 == 5
```

- Conditional execution

```
1 if x > 0 :  
2     print('x is positive')
```

- Catching exceptions

```
1 try:  
2     # Some code that throws an error  
3 except:  
4     # A useful error message
```

Conditional logic: R to Python

- The concept is the same, the Python syntax is simpler.
 - *In Python, pay attention to your indentation!*

R

```
1 if (x > 0) {  
2     print('x is positive')  
3 }
```

Python

```
1 if x > 0 :  
2     print('x is positive')
```

Tutorial #2: Iteration

- Iteration
- Core concepts:
 - Indefinite iteration (**while**)

```
1 while x < 10:  
2     # Do something with x
```

- Definite iteration (**for**)

```
1 for x in range(10):  
2     print(x)
```

Iteration: R to Python

- Just like `if` statements, the R concept is the same, the Python syntax is simpler.

R

```
1 for (x in 1:3) {  
2   print(x)  
3 }
```

Python

```
1 for x in [1, 2, 3]:  
2   print(x)
```

List comprehension

- Unique to Python: one-line iteration for lists.
 - Iterate over a list:

```
1 y = [x for x in [1, 2, 3]]
```

- Manipulate each element of a list:

```
1 [x + 3 for x in y]
```

Tutorial #3: Iteration

- More control flow tools §4.1-4.6
- Core concepts:
 - The `range()` function

```
1 range(3)
```

- Other control statements (`break`, `continue`, `pass`)
- `match` statements

```
1 match point:  
2     case 0:  
3         print("x is 0")  
4     case _:  
5         print(f"x is anything else!")
```

- Note: §4.6 mentions *functions* (`def`) which we will work