Control logic

Hamish Gibbs

Tutorial #1: Conditional logic

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

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

  - Conditional execution

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

  - Catching exceptions

```
try:
    # Some code that throws an error
except:
    # 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

```r
if (x > 0) {
    print('x is positive')
}
```

Python

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

Tutorial #2: Iteration

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

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

  - Definite iteration (`for`)

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

Iteration: R to Python

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

R

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

Python

```python
for x in [1, 2, 3]:
    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

```
range(3)
```

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

```
match x:
    case 0:
        print("x is 0")
    case _:
        print(f"x is anything else!")
```

Tutorial #3: Iteration

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