Naming Conventions in Python

Analytics Vidhya

Naming Conventions in Python

- Meaningful Identifiers

- Naming Variables and Functions

Meaningful Identifiers

```
def n_outLie(x, y='gaussian', n=3):

    if y == "gaussian":
        l = len(x[x < (x.mean()-(n*x.std()))])
        h = len(x[x > (x.mean()+(n*data.std()))])
        t = l+h
        return l, h, t

    elif y == 'whisker':
        l = len(x[x < x.quantile(0.25)-(1.5*(x.quantile(0.75) - x.quantile(0.25)))])
        h = len(x[x > x.quantile(0.75)+(1.5*(x.quantile(0.75)- x.quantile(0.25)))])
        t = l+h
        return l, h, t
```

Analytics Vidhya

Meaningful Identifiers

```
def num_outlier(data, criteria='gaussian', n=3):

    if criteria == "gaussian":
        low = len(data[data < (data.mean()-(n*data.std()))])
        high = len(data[data > (data.mean()+(n*data.std()))])
        tot = low+high
        return low, high, tot

    elif criteria == 'whisker':
        low = len(data[data < data.quantile(0.25)-(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        high = len(data[data > data.quantile(0.75)+(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        tot = low+high
        return low, high, tot
```

Analytics Vidhya

Meaningful Identifiers

```
def num_outlier(data, criteria='gaussian', n=3):

    if criteria == "gaussian":
        low = len(data[data < (data.mean()-(n*data.std()))])
        high = len(data[data > (data.mean()+(n*data.std()))])
        tot = low+high
        return low, high, tot

    elif criteria == 'whisker':
        low = len(data[data < data.quantile(0.25)-(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        high = len(data[data > data.quantile(0.75)+(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        tot = low+high
        return low, high, tot
```

Using relevant variable names

Analytics Vidhya

Meaningful Identifiers

```
def num_outlier(data, criteria='gaussian', n=3):

    if criteria == "gaussian":
        low = len(data[data < (data.mean()-(n*data.std()))])
        high = len(data[data > (data.mean()+(n*data.std()))])
        tot = low+high
        return low, high, tot

    elif criteria == 'whisker':
        low = len(data[data < data.quantile(0.25)-(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        high = len(data[data > data.quantile(0.75)+(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        tot = low+high
        return low, high, tot
```

Analytics Vidhya

Meaningful Identifiers

```
def num_outlier(data, criteria='gaussian', n=3):

    if criteria == "gaussian":
        low = len(data[data < (data.mean()-(n*data.std()))])
        high = len(data[data > (data.mean()+(n*data.std()))])
        tot = low+high
        return low, high, tot

    elif criteria == 'whisker':
        low = len(data[data < data.quantile(0.25)-(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        high = len(data[data > data.quantile(0.75)+(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        tot = low+high
        return low, high, tot
```

Numbers are never used by themselves

Analytics Vidhya

How to be meaningful?

How to be meaningful : Variable

- Characteristic or Property?

```
1  def slow_down(current_speed):
2      if current_speed > 5:
3          return current_speed - 5
4      else:
5          return 0
```

Analytics Vidhya

How to be meaningful : Variable

- Characteristic or Property?
- What is being Stored?

```
1  def factorial(value):
2      final_factorial = 1
3      for number in range(1,value+1):
4          final_factorial = final_factorial * number
5      return final_factorial
```

Analytics Vidhya

How to be meaningful : Variable

- Characteristic or Property?
- What is being Stored?
- Part of expression?

$$t = \frac{(x_1 - x_2)}{\sqrt{\frac{(s_1)^2}{n_1} + \frac{(s_2)^2}{n_2}}}$$

Analytics Vidhya

How to be meaningful : Variable

- Characteristic or Property?
- What is being Stored?
- Part of expression?

$$t = \frac{(x_1 - x_2)}{\sqrt{\frac{(s_1)^2}{n_1} + \frac{(s_2)^2}{n_2}}}$$

```
1 def t_statistic(mu1, mu2, sig1, sig2, n1, n2):
2     t_numerator = mu1 - mu2
3     t_denominator = ((s1**2)/n1 + (s2**2)/n2) ** 0.5
4     return t_numerator/t_denominator
```

Analytics Vidhya

How to be meaningful : Variable

- Characteristic or Property?
- What is being Stored?
- Part of expression?

$$t = \frac{(x_1 - x_2)}{\sqrt{\frac{(s_1)^2}{n_1} + \frac{(s_2)^2}{n_2}}}$$

```
1 def t_statistic(mu1, mu2, sig1, sig2, n1, n2):
2     t_numerator = mu1 - mu2
3     t_denominator = ((s1**2)/n1 + (s2**2)/n2) ** 0.5
4     return t_numerator/t_denominator
```

Analytics Vidhya

How to be meaningful : Variable

- Characteristic or Property?
- What is being Stored?
- Part of expression?

$$t = \frac{(x_1 - x_2)}{\sqrt{\frac{(s_1)^2}{n_1} + \frac{(s_2)^2}{n_2}}}$$

```
1  def t_statistic(mu1, mu2, sig1, sig2, n1, n2):
2      t_numerator = mu1 - mu2
3      t_denominator = ((s1**2)/n1 + (s2**2)/n2) ** 0.5
4      return t_numerator/t_denominator
```

Analytics Vidhya

How to be meaningful : Functions

How to be meaningful : Functions

- Verb : if mutating the inputs

How to be meaningful : Functions

- Verb : if mutating the inputs

```
def removing_gaussian_outliers(data, n_std):
    avg, std = data.mean(), data.std()
    return data[(data < (avg + (n_std*std))) & (data > (avg - (n_std*std)))]
```

Analytics Vidhya

How to be meaningful : Functions

- Verb : if mutating the inputs

```
def removing_gaussian_outliers(data, n_std):
    avg, std = data.mean(), data.std()
    return data[(data < (avg + (n_std*std))) & (data > (avg - (n_std*std)))]
```

Analytics Vidhya

How to be meaningful : Functions

- Noun : No mutation happens

How to be meaningful : Functions

- Noun : No mutation happens

```
def num_outlier(data, criteria='gaussian', n=3):

    if criteria == "gaussian":
        low = len(data[data < (data.mean()-(n*data.std()))])
        high = len(data[data > (data.mean()+(n*data.std()))])
        tot = low+high
        return low, high, tot

    elif criteria == 'whisker':
        low = len(data[data < data.quantile(0.25)-(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        high = len(data[data > data.quantile(0.75)+(1.5*(data.quantile(0.75)-data.quantile(0.25)))])
        tot = low+high
        return low, high, tot
```

Analytics Vidhya

Naming Functions and Variables

Naming Functions and Variables

- Variables and Functions are named similarly

Naming Functions and Variables

- Variables and Functions are named similarly
- ~~Function_name~~ , ~~VariableName~~ : Not recommended
- function_name , variableName : recommended

Naming Functions and Variables

- Variables and Functions are named similarly
- ~~Function_name~~ , ~~VariableName~~ : Not recommended
- function_name , variableName : recommended

Naming Functions and Variables

- Variables and Functions are named similarly
- ~~Function_name~~ , ~~VariableName~~ : Not recommended
- function_name , variableName : recommended

this_is_function_name

thisIsFunctionName

Analytics Vidhya

Naming Functions and Variables

- Variables and Functions are named similarly
- ~~Function_name~~ , ~~VariableName~~ : Not recommended
- function_name , variableName : recommended

this_is_function_name

thisIsFunctionName

Analytics Vidhya

Naming Functions and Variables

- Variables and Functions are named similarly
- ~~Function_name~~ , ~~VariableName~~ : Not recommended
- function_name , variableName : recommended
- Constants are declared using uppercase

```
PI = 3.14
```

Analytics Vidhya

Original Code

```
def n_outLie(x, y='gaussian', n=3):

    if y == "gaussian":
        l = len(x[x < (x.mean()-(n*x.std()))])
        h = len(x[x > (x.mean()+(n*data.std()))])
        t = l+h
        return l, h, t

    elif y == 'whisker':
        l = len(x[x < x.quantile(0.25)-(1.5*(x.quantile(0.75) - x.quantile(0.25)))])
        h = len(x[x > x.quantile(0.75)+(1.5*(x.quantile(0.75)- x.quantile(0.25)))])
        t = l+h
        return l, h, t
```

Analytics Vidhya

Result

```
def num_outliers(data, criteria='gaussian', n=3):

    if criteria == "gaussian":
        low = len(data[data < (data.mean()-(n*data.std()))])
        high = len(data[data > (data.mean()+(n*data.std()))])
        total = l+h
        return low, high, total

    elif criteria == 'whisker':
        M_FACTOR = 1.5
        QUART1 = 0.25
        QUART3 = 0.75
        low = len(data[data < data.quantile(QUART1)-(M_FACTOR*(data.quantile(QUART3) - data.quantile(QUART1)))])
        high = len(data[data > data.quantile(QUART3)+(M_FACTOR*(data.quantile(QUART3)- data.quantile(QUART1)))])
        total = low+high
        return low,high,total
```

Analytics Vidhya

Thank You