

Introduction to Biomedical Data Science and Health Informatics

Summary

Python for Data Science

Python for Data Science

- Arithmetic, logic, and string manipulation
 - +, -, *, /, **, and, or, not, ==, !=
 - items = some_string.split()
- Data types
 - Lists
 - Dictionaries
 - Sets
- Structure
 - def
 - if
 - else
- Vast library of functions available
 - import ...
 - from ... import ...

Regular expressions

- import re
- Looks for patterns in text.

```
re.findall(r"([a-zA-Z]+) is ([0-9]+) years old", note)  
[('John', '32'), ('Sarah', '29'), ('Sam', '2')]
```

Data Frames

```
import pandas as pd

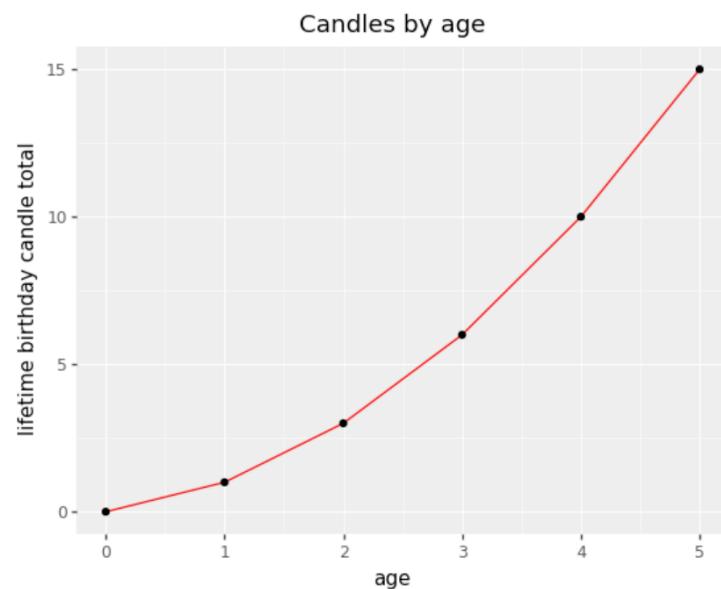
data = pd.read_csv('filename.csv')
age = data['age']

seniors = data[data['age'] >= 65]

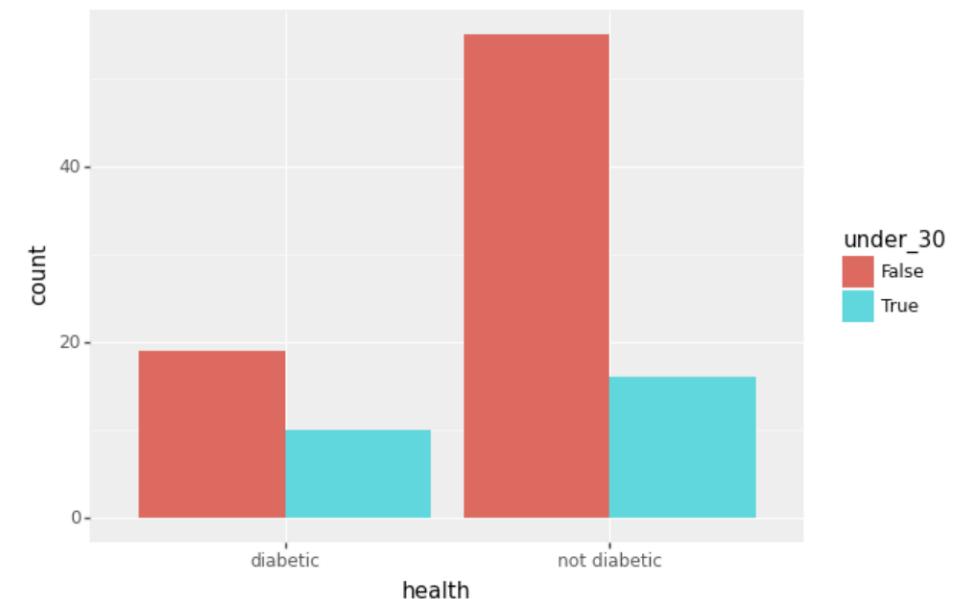
has_alzheimers = data['conditions'].str.contains("Alzheimer's")
```

import plotnine as p9

```
(  
    p9.ggplot(data, p9.aes(x='age', y='lifetime_bday_candles'))  
    + p9.geom_line(color='red')  
    + p9.geom_point()  
    + p9.labs(y='lifetime birthday candle total', title='Candles by age')  
)
```



```
patients['under_30'] = patients['ages'] < 30  
(  
    p9.ggplot(patients, p9.aes(x='health', fill='under_30'))  
    + p9.geom_bar(position='dodge')  
)
```



Relational Databases

Structured Query Language (SQL)

```
pd.read_sql(''  
    SELECT thing1, thing2  
    FROM table1_name  
        INNER JOIN table2_name  
            ON table1_name.column_name1 = table2_name.column_name2  
    WHERE logical_constraint  
    GROUP BY column_name  
    ORDER BY column_name ASC/DESC  
'', conn)
```

Data Cleaning and Visualization

Exploratory data analysis

- Always your first step.
- Focus on understanding the data not testing the hypothesis.
- Look at the data.
- Get summary data.
 - `.describe()`
 - `.value_counts()`
- Univariate and bivariate analyses.
 - Distribution, outliers.

Wickham's A layered grammar of graphics

- A plot consists of several components added together:
 - Data and aesthetic mappings (`p9.ggplot`, `p9.aes`)
 - Geometric objects (`p9.geom_bar`, `p9.geom_line`, `p9.geom_point`, ...)
 - Scales (e.g. `p9.scale_y_continuous(trans='log10')`)
 - Facet specification (e.g. `p9.facet_wrap('year')`, `p9.facet_grid('var1 ~ var2')`)
 - Statistical transformations (e.g. `stat='bin'`, `stat='identity'`)
 - Coordinate system (e.g. `p9.coord_flip()`, `p9.coord_fixed(ratio=1)`)

Data cleaning

- Non-standardized data
- Inconveniently structured data
 - Tidy-ing datta
 - Data with multiple factors
- Duplicate data
- Incorrect values
- Missing values

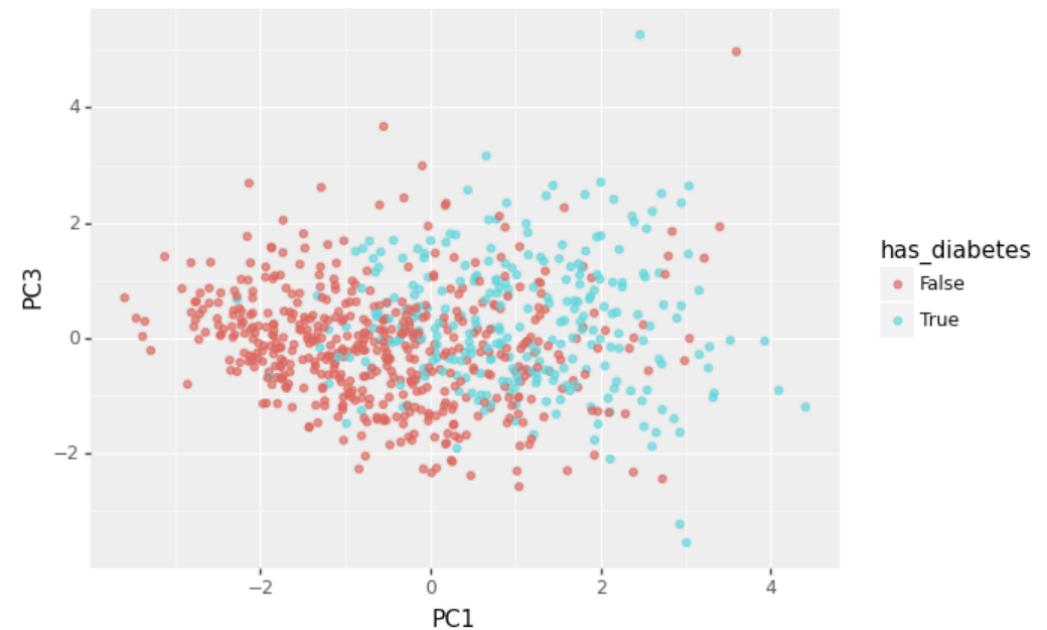
Machine Learning and Bioinformatics

Principal Component Analysis (PCA)

- Dimensionality reduction technique.
- Rotation of the dataset.

```
from sklearn import decomposition

pca = decomposition.PCA(n_components=4)
X_reduced = pd.DataFrame(
    pca.fit_transform(X),
    columns=['PC1', 'PC2', 'PC3', 'PC4']
)
```



sklearn

- Module for machine learning.
- Training, cross-validation, testing for several algorithms.
 - Logistic regression
 - Random forest
 - Neural networks
 - more

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix

clf = RandomForestClassifier(
    n_estimators=some_number,
    max_depth=D,
    random_state=0)

sc = cross_val_score(clf, X_train, y_train, cv=10)

clf.fit(X_train, y_train)
y_predicted = clf.predict(X_test)

cm = confusion_matrix(y_predicted, y_test)
```

Natural Language Processing

nltk module

- Tokenization and normalization
- Stop word removal
- Part of speech tagging (POS)
- Named entity recognition (NER)
- Stemming and lemmatization