INFO370 Lab 8: Simple regression
Deadline: Wed, Nov 27th, 5pm

Your name:

November 22, 2019

Instructions

This lab asks you to do a) multiple regression using existing packages; and b) implement it from scratch using linear algebra. While the first is what you normally do, the second prepares you to more complex methods where you have to manipulate the data in matrices.

1 Boston housing data

Your are using Boston housing data, a famous dataset about house prices across 506 Boston neighborhoods in 1970-s. Your task is to model the price (*medv*) as a function of the other variables. You can consult https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html for more information.

1. Load the data. In R, it is built into MASS package so you can access it just like

```
head(MASS::Boston, 3)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8
##    black lstat medv
## 1 396.90  4.98 24.0
## 2 396.90  9.14 21.6
## 3 392.83  4.03 34.7
```

In python, it is in `sklearn` package and you can access it as

```
from sklearn.datasets import load_boston
boston = load_boston()
boston.data[:3,:]
```

```
## array([[6.3200e-03, 1.8000e+01, 2.3100e+00, 0.0000e+00, 5.3800e-01,
##         6.5750e+00, 6.5200e+01, 4.0900e+00, 1.0000e+00, 2.9600e+02,
##         1.5300e+01, 3.9690e+02, 4.9800e+00],
##        [2.7310e-02, 0.0000e+00, 7.0700e+00, 0.0000e+00, 4.6900e-01,
##         6.4210e+00, 7.8900e+01, 4.9671e+00, 2.0000e+00, 2.4200e+02,
##         1.7800e+01, 3.9690e+02, 9.1400e+00],
```

```
##         [2.7290e-02, 0.0000e+00, 7.0700e+00, 0.0000e+00, 4.6900e-01,
##          7.1850e+00, 6.1100e+01, 4.9671e+00, 2.0000e+00, 2.4200e+02,
##          1.7800e+01, 3.9283e+02, 4.0300e+00]])

boston.target[:3]

## array([24. , 21.6, 34.7])
```

As you can see, in R it is a data frame and you have to convert it into a matrix (you can use `as.matrix`). In python it is already a matrix, separated for predictors (data) and outcome (target).

1.1 Multiple regression

Next, you use an existing regression package to estimate the model

$$\text{medv}_i = \boldsymbol{\beta}^\intercal \mathbf{x}_i + \epsilon_i \tag{1}$$

where *medv* is the median house value in neighborhood i, and $\mathbf{x}$ are all the other descriptors in the data.

In R you can just use 'lm' as we have done in class (note: there is a way to tell "include all variables"), in python there are two options: one easy but harder to interpret, the other harder but easy to interpret. Here are suggestions for the 2nd way:

First, transform it into a dataframe:

```
import pandas as pd
data = pd.DataFrame(boston.data, columns=boston.feature_names)
data['medv'] = boston.target
data.head(3)
```

```
##       CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  PTRATIO       B  LSTAT  m
## 0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0     15.3  396.90   4.98  2
## 1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0     17.8  396.90   9.14  2
## 2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0     17.8  392.83   4.03  3
```

and thereafter use statsmodels formula api (see https://www.statsmodels.org/dev/example_formulas.html).

1. Estimate the regression model and interpret the output.
2. Interpret a single result: how does the price depend on the average size (RM = average number of rooms)?

1.2 Use the matrix formula

Next, let's use the matrix formula:

$$\boldsymbol{\beta}^* = (\mathsf{X}^\intercal \cdot \mathsf{X})^{-1} \cdot \mathsf{X}^\intercal \cdot \mathbf{y} \tag{2}$$

where X is your data matrix, $^\intercal$ is the transposition operator, and $\mathbf{y}$ is your outcome vector.

1. Convert your data into the design matrix X. Note: in R you start with a data frame.

   Hint: don't forget the column of ones!

2. compute the optimal $\boldsymbol{\beta}^*$ using the formula above.

3. compare the results you get with the regression results above. What do you find?

4. Why do you need the constant column? What happens if you leave it out? Try this! What do you find?