CIS 124: Intro to Data Science

Miles Kent

Data Visualization, Causality and Experiments
Python Expression, Strings and Table methods
Charts and Histograms
Probability and Iteration, Sampling, Simulation
Hypothesis Testing and Models
Data interpretation related to the above

1 Section 2

Observation vs Experiment
What constitutes causation?

2 Section 3

Basic Python & its conventions

2.1 Table().select()

tbl.select("Col1", "Col2", ...)

2.2 Table().sort()

tbl.sort("Col", descending=**BOOL**)
descending = True → big to small

3 Section 4

3.1 make_array(...)

Not really a vector. Element wise operations, i.e. when you multiply two 1D arrays it doesn't dot them, it multiplies each corresponding element and gives the resultant array. It's more of a table column except it's not in a Table() yet.
make_array(1, 2, 3, 4)

3.2 Table().column(...)

Table().column(nameorindex)
Outputs array

4 Section 5

Table().num_rows
Table().num_columns
Table().labels

4.1 Table().relabled(...)

Table().relabeled("col1name", "NEWNAME")

4.2 Table.read_table(...)

Table.read_table("filename.csv")

4.3 Table().with_columns(...)

Table().with_columns("col1name", col1arr, ...)

4.4 Table().drop(...)

Table().drop("col1name", ...)

4.5 Table().take(...)

Table().take(rowindicestouse)

4.6 Table().where(...)

Table().take("colname", condition)
condition uses "are" syntax.
are.equal_to()
are.above()

are.above_or_equal_to()
are.below()
are.below_or_equal_to()
are.between() [A, B)
are.between_or_equal_to() [A, B)
are.contained_in()
are.containing()
are.strictly_between() (A, B)

5 Section 6

5.1 Table().barh(...)

Table().barh(yaxiscategoriescolumn) Table().barh(yaxiscategoriescolumn, values)

5.2 Table().show(...)

Table().show(num)

6 Section 7

6.1 Table().scatter(...)

Table().scatter(xcol, ycol, fit_line=BOOL)

6.2 Table.plot(...)

Table().plot(xcol, ycol)
Similar to scatter but it connects the dots

7 Section 8

7.1 Table().hist(...)

Table().hist("colname", bins=make_array(...))

8 Section 9

8.1 Table().apply(...)

Table().apply(functionname, "columnname")

9 Section 10

9.1 Table().group(...)

Table().group(columnname(s))
Table().group(columnname(s), func)
Rows are grouped by unique values. Func would be something that takes an arrat and returns a single value.

9.2 Table.pivot(...)

Table.pivot("col1name", "col2name", ...)
Table.pivot("col1name", "col2name", ..., values="colXname", collect=func)
Col1 is x categories, Col2 is y categories, values is the data points in each box, collect is the function that is applied to each categorical data point, e.g. collect=sum would give you the sum for each one.

10 Section 11

10.1 Table().join(...)

tbl1.join("colthingfrom1", tbl2, "colsamethingfrom2") Joins two tables based on a like category/column.

11 Section 12

Yawn...

12 Section 13

Loops and also you can iterate over make_array(...) btw.

13 Section 14

P(Both A and B) = P(A) * P(B) if independent
If A can only happen in one of two possibilities, P(A) = P(A1) + P(A2)

14 Section 15

Deterministic vs Probability vs Uniform Random Sample
$textbf Deterministic$: you just choose some values for a sample. For example, I choose 3, 18, and 100 for my sample.
$textbf Probability$: a sample for which you can calculate the probability of any subset being the sample before the sample has been drawn.
$textbf UniformRandom$: randomly chosen.
Also understand the law of large numbers.
Whatever vairation distance is.

15 Section 16-18

Stats stuff.

16 Section 19

16.1 Table().sample(...)

Table.sample(nrows, with_replacement=BOOL)