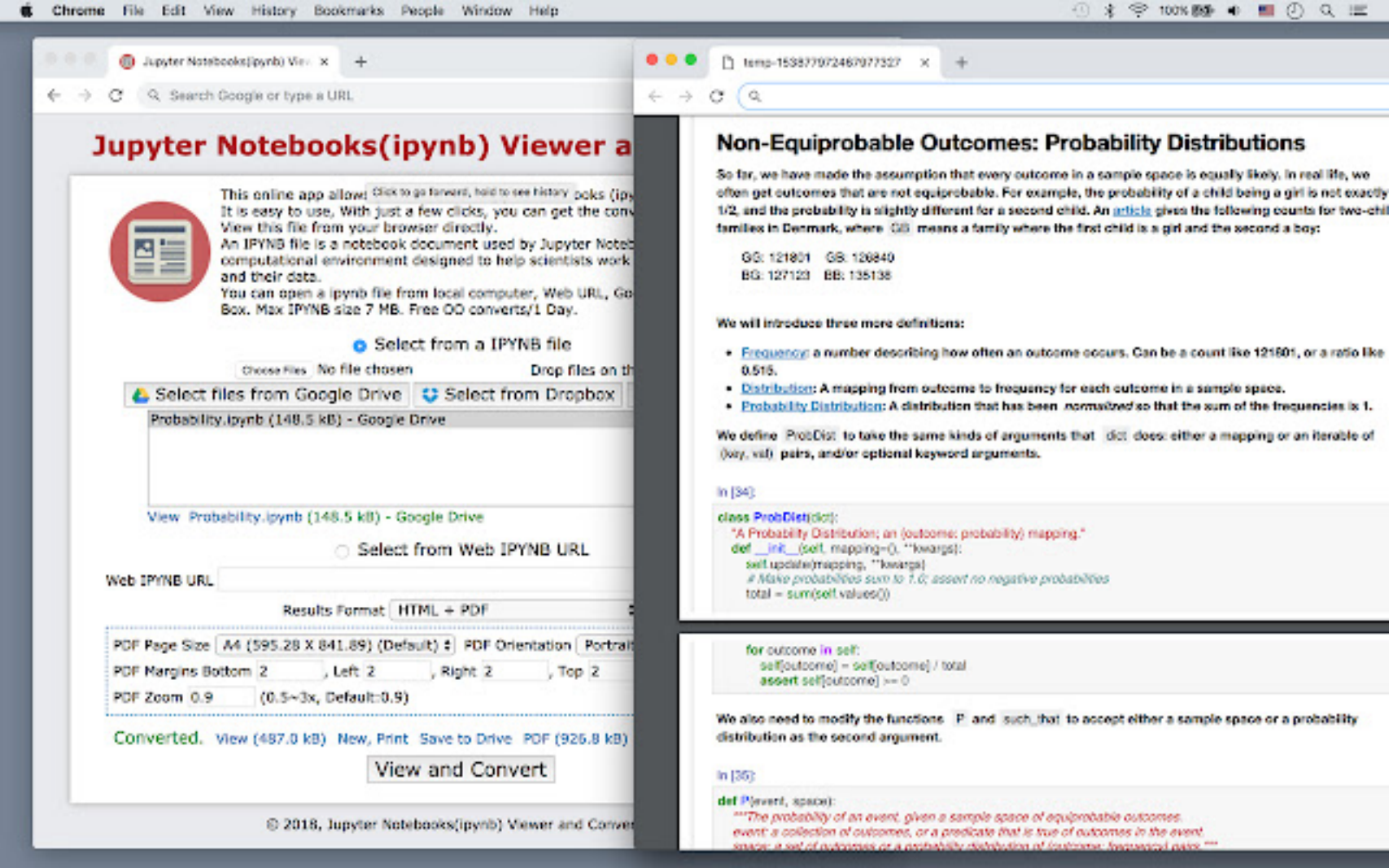
Chrome File Edit View History Bookmarks People Window Help
Jupyter Notebooks(ipynb) Viewer
Search Google or type a URL
Jupyter Notebooks(ipynb) Viewer a
This online app allow
It is easy to use, With just a few clicks, you can get the
View this file from your browser directly.
An IPYNB file is a notebook document used by Jupyter
computational environment designed to help scientists work
and their data.
You can open a ipynb file from local computer, Web URL,
Box. Max IPYNB size 7 MB. Free 00 converts/1 Day.
Select from a IPYNB file
Choose Files No file chosen
Drop files on
Select files from Google Drive
Select from Dropbox
Probability.ipynb (148.5 kB) - Google Drive
View Probability.ipynb (148.5 kB) - Google Drive
Select from Web IPYNB URL
Web IPYNB URL
Results Format HTML + PDF
PDF Page Size A4 (595.28 X 841.89) (Default)
PDF Orientation
PDF Margins Bottom 2 , Left 2 , Right 2 , Top 2
PDF Zoom 0.9 (0.5~3x, Default:0.9)
Converted. View (487.0 kB) New, Print Save to Drive PDF (926.8 kB)
View and Convert
© 2018, Jupyter Notebooks(ipynb) Viewer and
Non-Equiprobable Outcomes: Probability Distributions
So far, we have made the assumption that every outcome in a sample space is equally likely. In real life, we often get outcomes that are not equiprobable. For example, the probability of a child being a girl is not exactly 1/2, and the probability is slightly different for a second child. An article gives the following counts for two-child families in Denmark, where GB means a family where the first child is a girl and the second a boy:
GG: 121801 GB: 126840
BG: 127123 BB: 135138
We will introduce three more definitions:
Frequency: a number describing how often an outcome occurs. Can be a count like 121601, or a ratio like 0.515.
Distribution: A mapping from outcome to frequency for each outcome in a sample space.
Probability Distribution: A distribution that has been normalized so that the sum of the frequencies is 1.
We define ProbDist to take the same kinds of arguments that dict does: either a mapping or an iterable of (key, val) pairs, and/or optional keyword arguments.
In [34]:
class ProbDist(dict):
"A Probability Distribution; an {outcome: probability} mapping."
def __init__(self, mapping=(), **kwargs):
self.update(mapping, **kwargs)
# Make probabilities sum to 1.0; assert no negative probabilities
total = sum(self.values())
for outcome in self:
self[outcome] = self[outcome] / total
assert self[outcome] >= 0
We also need to modify the functions P and such_that to accept either a sample space or a probability distribution as the second argument.
In [35]:
def P(event, space):
"""The probability of an event, given a sample space of equiprobable outcomes.
event: a collection of outcomes, or a predicate that is true of outcomes in the event.
space: a set of outcomes or a probability distribution of outcomes. Immigrant pairs."""