AIMS DMG Sage Demonstration 08
AIMS 2013-14: Designs, Matroids and Graphs

Rob Beezer
University of Puget Sound

Nancy Neudauer
Pacific University

January 8, 2014

1 Matroids in Sage

Constructors. How do we make matroids in Sage? Much the same way we make anything in Sage, with a "constructor". A function that creates the object we wish to have. In this case we use `Matroid()` which is quite intelligent about handling the input it receives.

We are going to build a binary matroid, the points are column vectors with entries from $GF(2)$, which we collect for convenience in a matrix.

```
mat = Matrix(GF(2), [
[1,0,0,0,1,1,0,1],
[0,1,0,0,1,0,1,1],
[0,0,1,0,0,1,1,1],
[0,0,0,1,1,1,1,1]
])
mat
```

```
M = Matroid(mat)
M
```

Let's run down the most basic properties of a matroid for this example.

```
M.groundset()
```

```
M.groundset_list()
```

The bases would be next.

```
M.bases()
```

What is an iterator? A useful way to avoid a "combinatorial" explosion!

```
bit = M.bases()
```

```
for b in bit:
   print b
```

```
list(bit)
```

How about a list of bases without the "frozenset" stuff? Use a list comprehension and covert the sets to lists.

```
[list(b) for b in bit]
```

The rank of the matroid is obvious now, but Sage knows it too.

```
M.rank()
```

The circuits — the minimally dependent sets. An iterator again. Check a few of these back against the original matrix.

```
cit = M.circuits()
[list(c) for c in cit]
```

Even faster, in one line, but as frozen sets.

```
list(M.circuits())
```

We can find all the independent sets of a given size.

```
M.independent_r_sets(r=2)
```

```
M.independent_r_sets(r=5)
```

2 More Matroids

We just built a matroid from scratch, Sage has many more available. Use tab-completion on `matroids.` or `matroids.named_matroids.`.

```
matroids.
```

```
matroids.named_matroids.
```