Impact of CERN Open Hardware Study (v.0.6)

June 21, 2017

1 Impact of CERN Open Hardware

This notebook contains the analysis for the preliminary CERN Open Hardware (OH) survey conducted by the CERN Knowledge Transfer (KT) group in 2016.

1.1 Methodology

In total, 149 cases were collected with an online survey for the "Impact of CERN Open Hardware study" and, after data clean-up of replicates and empty responses, the final dataset contained 146 respondents. For the analytic purposes, basic descriptive statistics were generated. Questions of evaluation and perception were not quantified but visualised using a Likert scale.

In addition to the survey, 14 face-to-face interviews were conducted with Open Hardware experts within and outside CERN. The interview material was first transcribed and then analysed to identify overarching themes. The qualitative analysis was conducted by comparing and contrasting annotations between the two authors to elucidate what each one read in the qualitative corpus. For the analysis we present in the next section, a final pass on the qualitative material was performed to refine the list of keywords we generated. A matrix of "theme co-occurrence" was used to facilitate the identification of key controversies. This matrix is described below.

There are caveats with respect to the scope of the "Impact of CERN Open Hardware study." First, the survey was meant as an exploratory device of limited depth and breadth, which is not to be taken as representative of the broader OH community. Since the goal was to learn initially about the basic profile and practical experiences of community members around CERN, target groups were initially categorised according to their self-declared role as "supporters," "procurement," "legal & administrative personnel," and "firms." Key CERN OH supporters were interviewed at first and several contacts were obtained. Interviewees were also identified through the OSHWA mailing-list after the survey was collected by CERN-KT.

```
In [1]: import pandas as pd
        import seaborn as plt
```

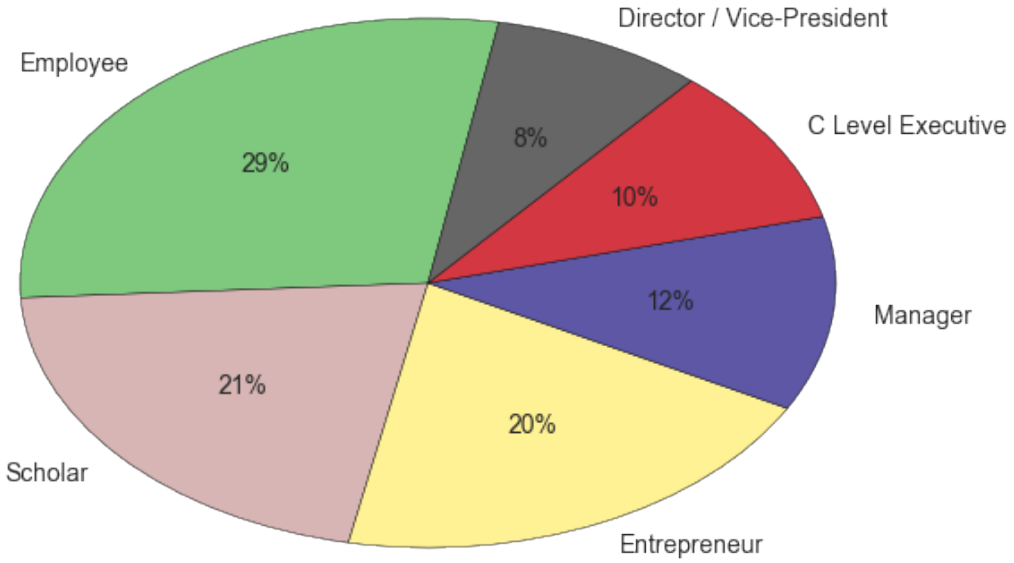
```
In [2]: dt = pd.read_csv('survey_data.csv', sep=';', header=0)
        qualicodes = pd.read_csv('code_co-ocurrence.csv', header=0, index_col=0)
```

1.2 Survey Questions

1.2.1 Which of the following most closely matches your job title?

```
In [3]: % matplotlib inline
        jobtitle = dt['job title'] # .dropna() or, fillna('No response')
```

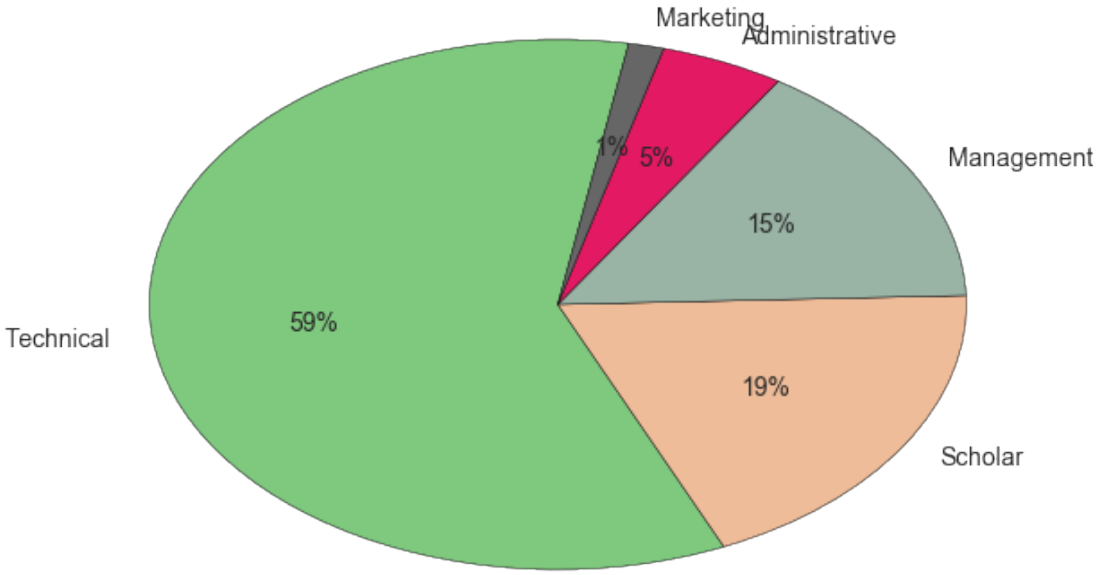
```
job_labels = ['Employee', 'Scholar', 'Entrepreneur', 'Manager', \
              'C Level Executive', 'Director / Vice-President']
fig = jobtitle.value_counts().plot(kind='pie', labels=job_labels, autopct='
                                   startangle=80, colormap='Accent')
fig.set_ylabel('')
fig.figure.tight_layout()
fig.figure.savefig('/tmp/job_title.png', dpi=300)

# Bar graph
#fig = jobtitle.value_counts().sort_values().plot(kind='barh', alpha=0.5, c
#fig.set_yticklabels(['Employee', 'Scholar', 'Entrepreneur', 'Manager', \
#                     'C Level Executive', 'Director / Vice-President']
```



1.2.2 In which function do you work?

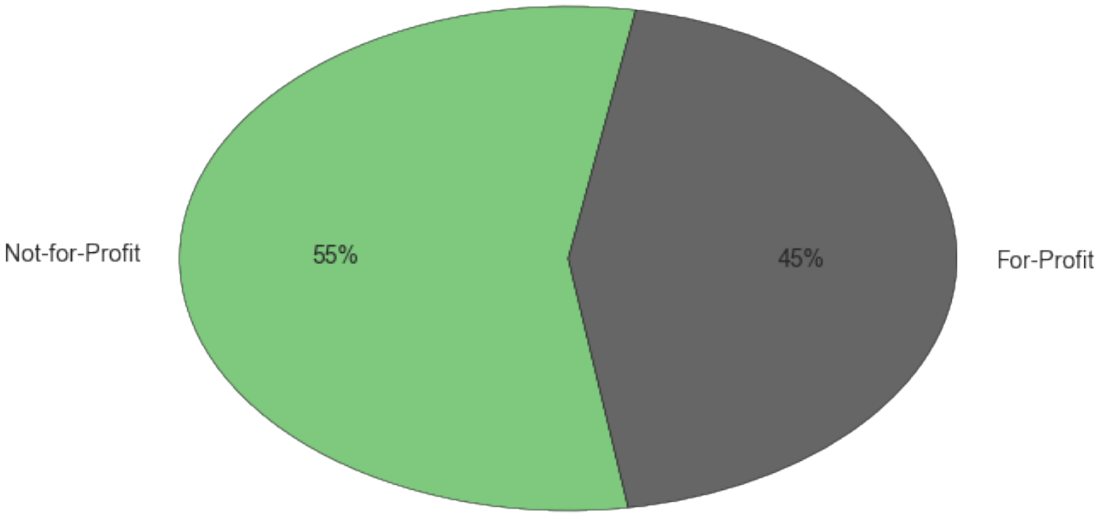
In [4]:
```
jobtitle = dt['job function']
job_titles = ['Technical', 'Scholar', 'Management', 'Administrative', 'Mark
fig = jobtitle.value_counts().plot(kind='pie', labels=job_titles, autopct='
                                   startangle=80, colormap='Accent')
fig.set_ylabel('')
fig.figure.tight_layout()
fig.figure.savefig('/tmp/function_work.png', dpi=300)

# Bar graph
#fig = jobtitle.value_counts().sort_values().plot(kind='barh', alpha=0.5, c
```

```
        #fig.set_yticklabels(['Marketing', 'Administrative', 'Management', \
        #                      'Scholar', 'Technical'])
```



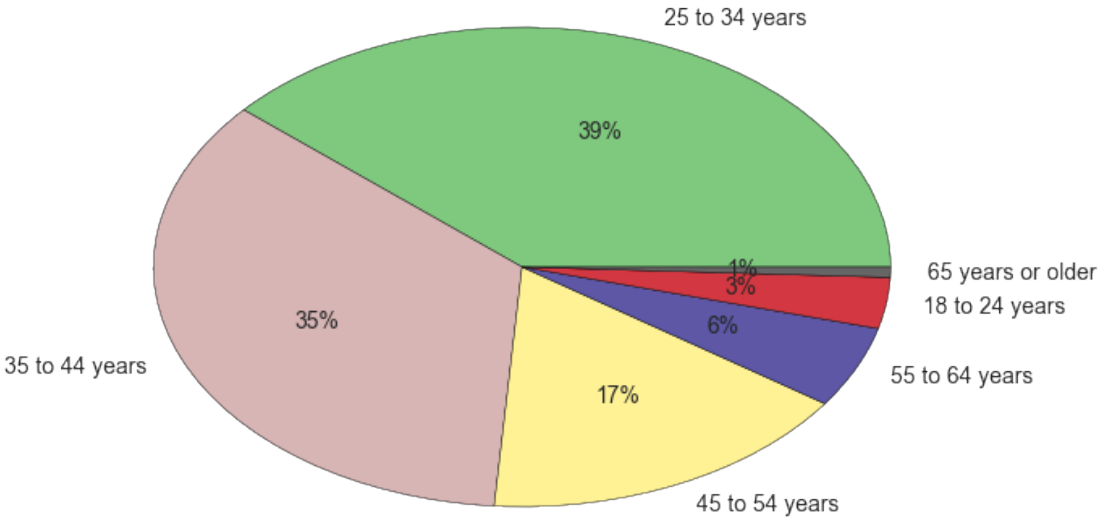
1.2.3 In which sector does your organisation operate?

```
In [5]: orgsector = dt['sector of organization']
        pie_labels = ['Not-for-Profit', 'For-Profit']
        fig = orgsector.value_counts().plot(kind='pie', labels=pie_labels, autopct=
                                            startangle=80, colormap='Accent')
        fig.set_ylabel('')
        fig.figure.tight_layout()
        fig.figure.savefig('/tmp/org_sector.png', dpi=300)
```



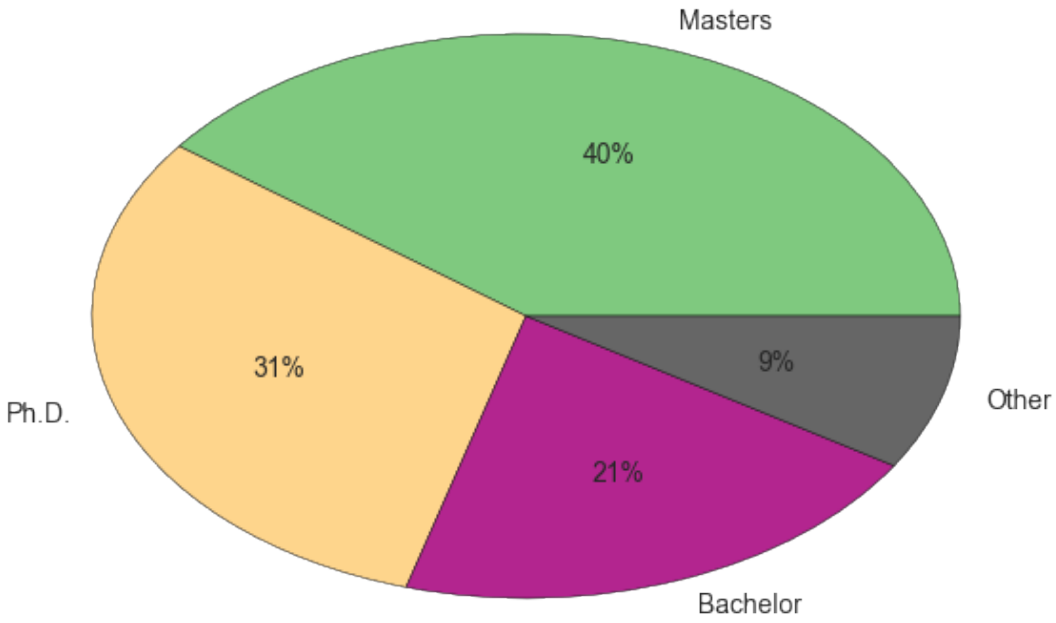
1.2.4 What is your age?

```
In [6]: age = dt['age']
        pie_labels = ['25 to 34 years', '35 to 44 years', '45 to 54 years', '55 to
                      '18 to 24 years', '65 years or older']
        fig = age.value_counts().plot(kind='pie', labels=pie_labels, autopct='%.0f%
        fig.set_ylabel('')
        fig.figure.tight_layout()
        fig.figure.savefig('/tmp/age_groups.png', dpi=300)

        # Alternative viz, horizontal bars
        #fig = jobtitle.value_counts().plot(kind='barh', alpha=0.5)
        #fig.set_yticklabels(['25 to 34 years', '35 to 44 years', '45 to 54 years',
        #                     '18 to 24 years', '65 years or older'])
```



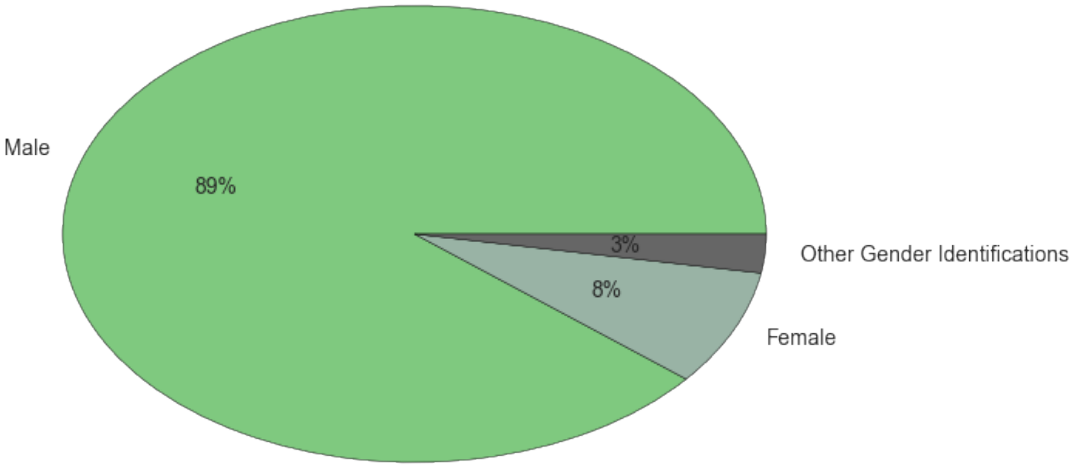
1.2.5 Education Levels

```
In [7]: education = dt['education level']
        pie_labels = ['Masters', 'Ph.D.', 'Bachelor','Other']
        fig = education.value_counts().plot(kind='pie', labels=pie_labels, autopct=
                                            colormap='Accent')
        fig.set_ylabel('')
        fig.figure.tight_layout()
        fig.figure.savefig('/tmp/education_levels.png', dpi=300)
```



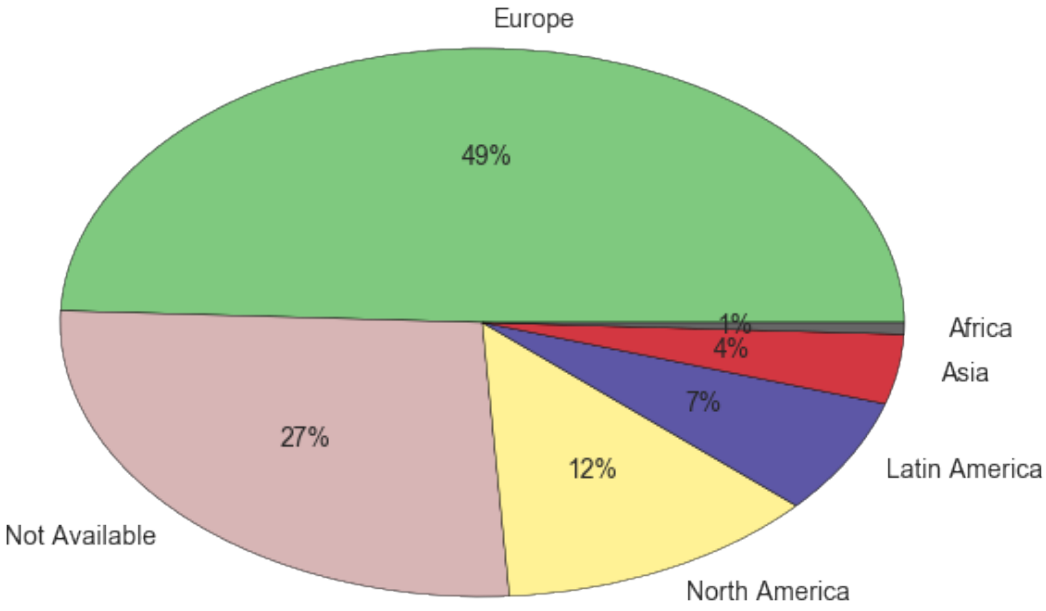
1.2.6 Gender

```
In [8]: gender = dt['gender']
        pie_labels = ['Male', 'Female', 'Other Gender Identifications']
        fig = gender.value_counts().plot(kind='pie', labels=pie_labels, autopct='%.
                                         colormap='Accent')
        fig.set_ylabel('')
        fig.figure.tight_layout()
        fig.figure.savefig('/tmp/gender.png', dpi=300)
```



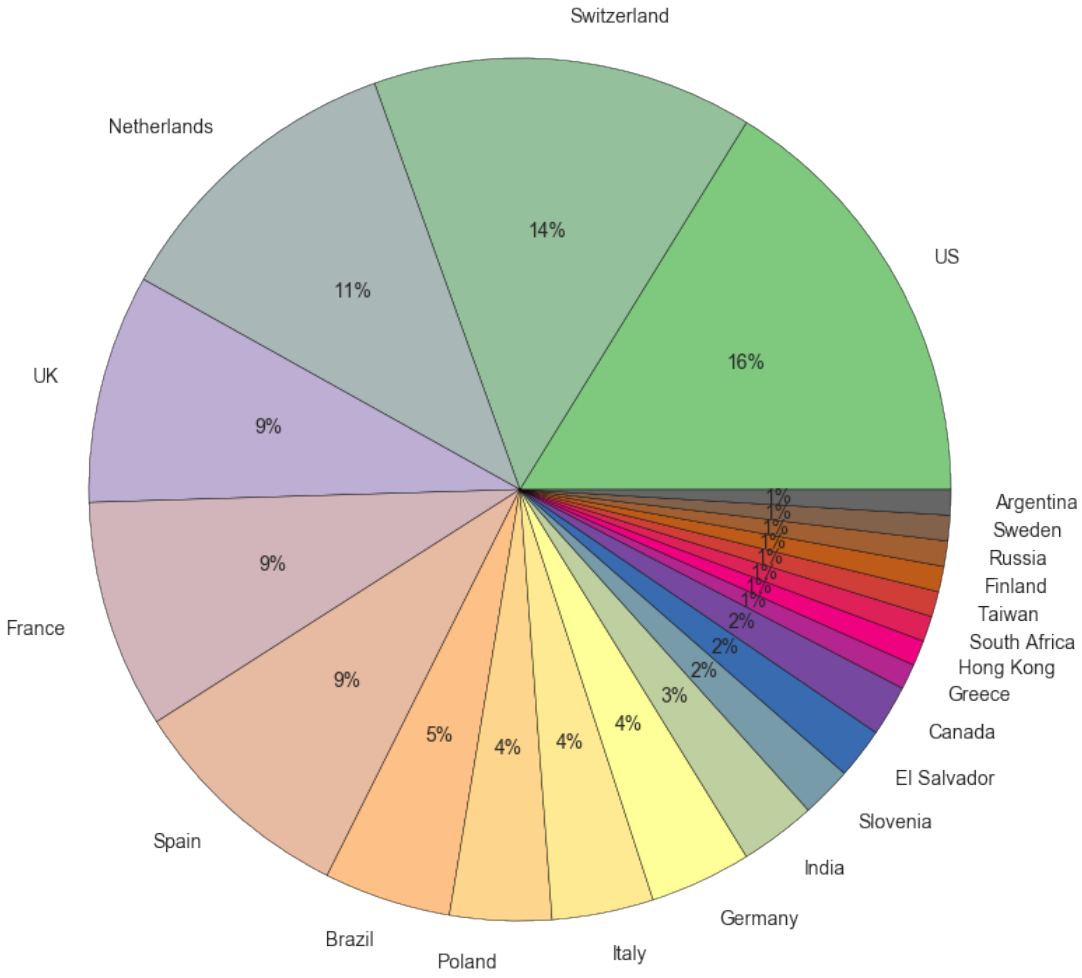
1.2.7 Region

```
In [9]: #location = dt[['Location', 'Country']]
        location = dt['Location']
        location_labels = ['Europe', 'Not Available', 'North America', \
                           'Latin America', 'Asia', 'Africa']

        fig = location.value_counts().plot(kind='pie', labels=location_labels, auto
                                           colormap='Accent')
        fig.set_ylabel('')
        fig.figure.tight_layout()
        fig.figure.savefig('/tmp/location.png', dpi=300)
```



1.2.8 Country

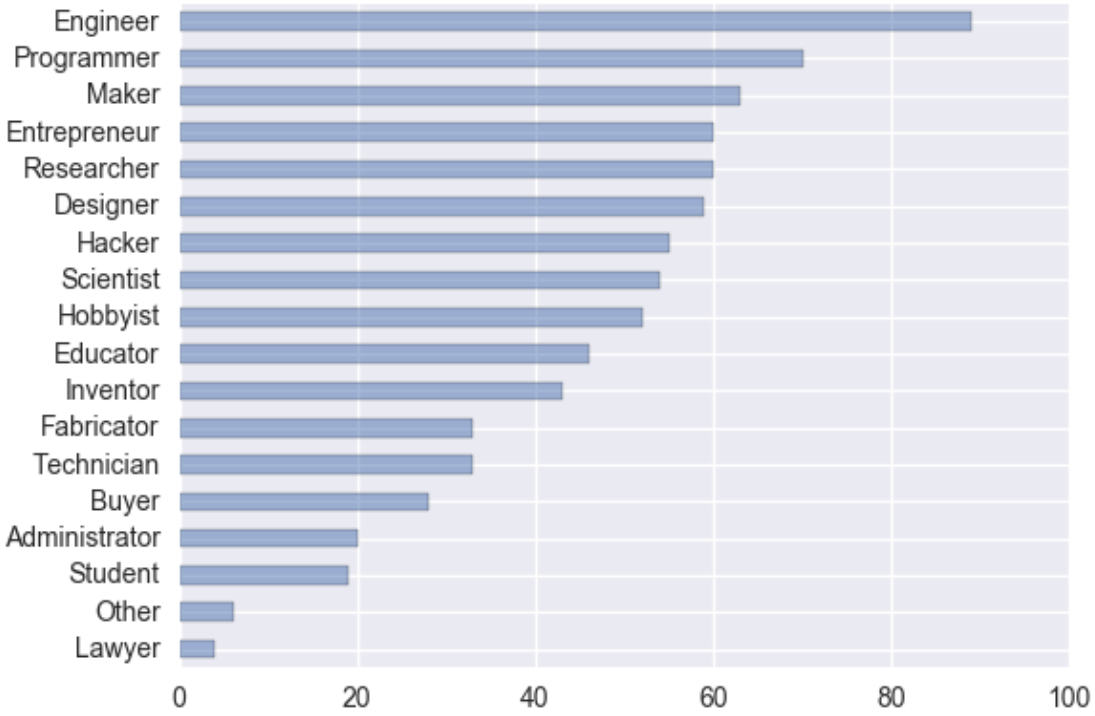
```
In [10]: country = dt['Country'].value_counts().dropna()
         fig = country.plot(kind='pie', autopct='%.0f%%', figsize=(8, 8), colormap=
         fig.set_ylabel('')
         fig.figure.tight_layout()
         fig.figure.savefig('/tmp/country.png', dpi=300)
```



1.2.9 Role in the OSHW Community

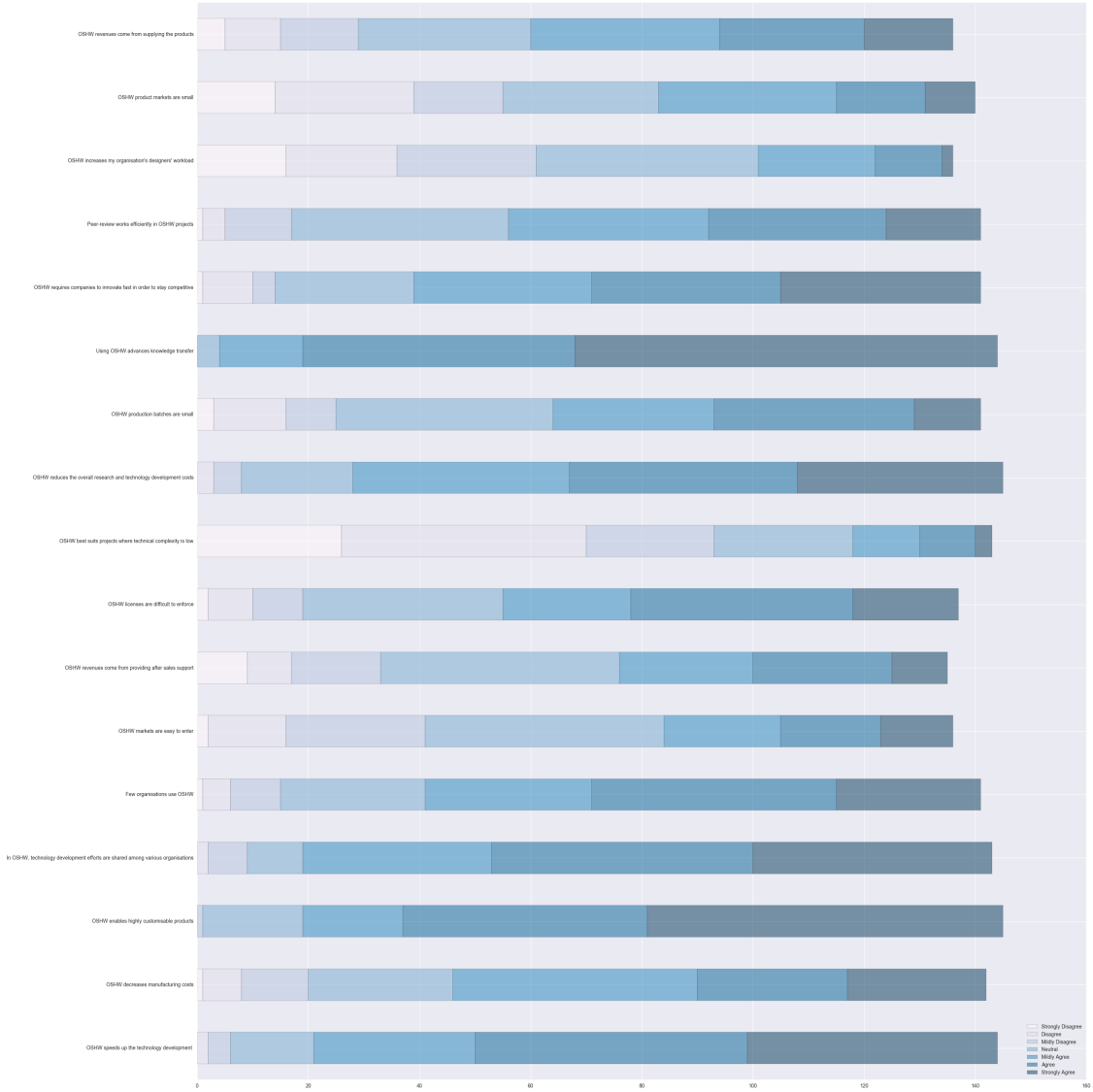
```
In [11]: # role_list = {1: 'Student', 2: 'Hobbyist', 3: 'Entrepreneur', 4: 'Technic
         #          5: 'Educator', 6: 'Scientist', 7: 'Fabricator', 8: 'Engineer', \
         #          9: 'Programmer', 10: 'Maker', 11: 'Hacker', 12: 'Designer', \
         #          13: 'Inventor', 14: 'Researcher', 15: 'Administrator', \
         #          16: 'Lawyer', 17: 'Buyer', 18: 'Other'}

         roles = dt['role']
         roles_count = roles.str.split(',').apply(lambda x: pd.Series(x).value_cour
         roles_count.sort_values(inplace=True)
         plt.set_style('darkgrid')
         fig = roles_count.plot(kind='barh', alpha=0.5, xlim=(0,100))
```

```
        fig.set_yticklabels(['Lawyer', 'Other', 'Student', 'Administrator', 'Buyer
                             'Technician', 'Fabricator', 'Inventor', 'Educator', '
                             'Scientist', 'Hacker', 'Designer', 'Researcher', 'Ent
                             'Maker', 'Programmer', 'Engineer'])

        fig.figure.tight_layout()
        fig.figure.savefig('/tmp/roles.png', dpi=300)
```



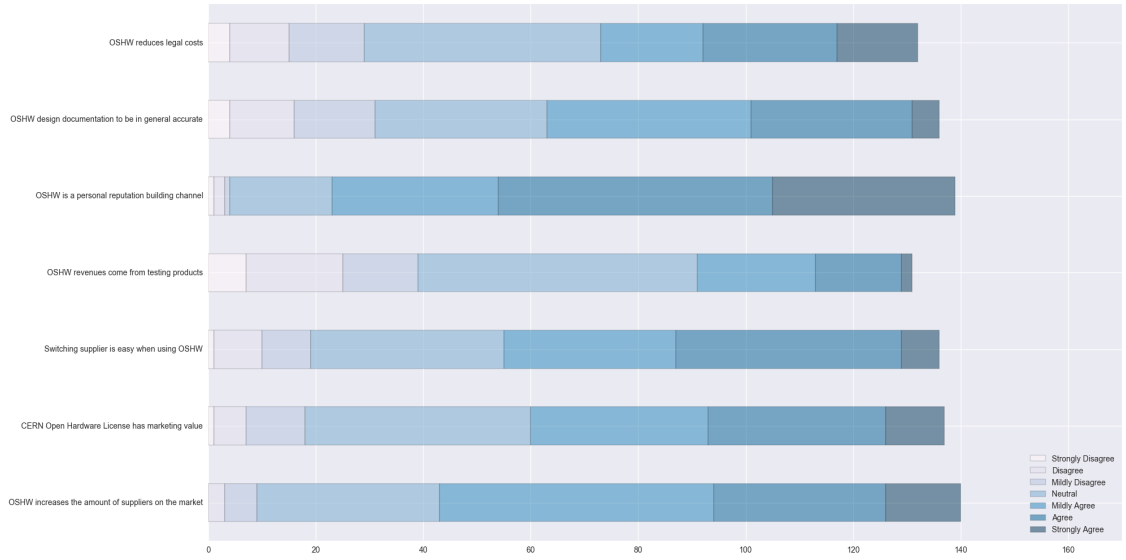
1.2.10 Community Perception, Part 1 (questions 9-22)

```
In [12]: perceptions = dt.ix[:,9:26]
         results = perceptions.apply(pd.value_counts).fillna(0).T
         fig = results.plot(kind='barh', stacked=True, alpha=0.5, colormap='PuBu',
                            xlim=(0,160))
         fig.legend(('Strongly Disagree','Disagree', 'Mildly Disagree', 'Neutral',
                     'Mildly Agree', 'Agree', 'Strongly Agree'), loc='lower right')

         fig.figure.tight_layout()
         fig.figure.savefig('/tmp/comm_perceptions_1.png', dpi=300)
```



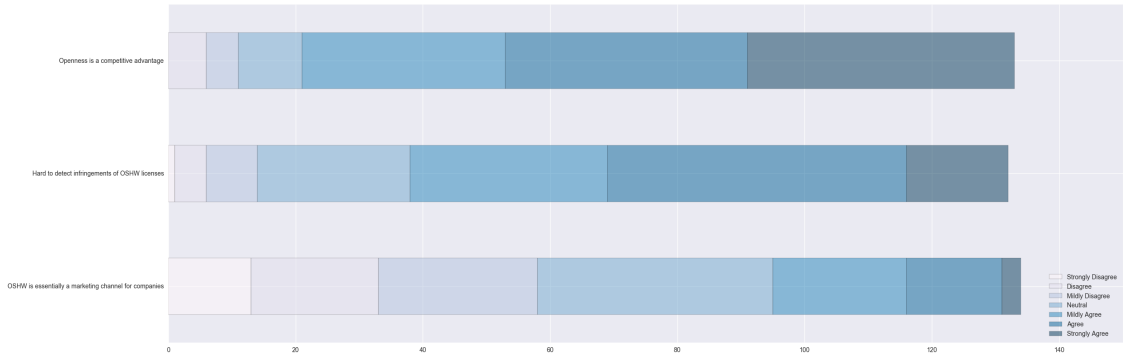
1.2.11 Community Perception, Part 2 (questions 28-35)

```
In [13]: perceptions2 = dt.ix[:,29:36]
         results = perceptions2.apply(pd.value_counts).fillna(0).T
         fig = results.plot(kind='barh', stacked=True, alpha=0.5, colormap='PuBu',
                            xlim=(0,170))
         fig.legend(('Strongly Disagree','Disagree', 'Mildly Disagree', 'Neutral',
                     'Mildly Agree', 'Agree', 'Strongly Agree'), loc='lower right')

         fig.figure.tight_layout()
         fig.figure.savefig('/tmp/comm_perceptions2.png', dpi=300)
```



1.2.12 Community Perception, Part 3 (questions 37-39)

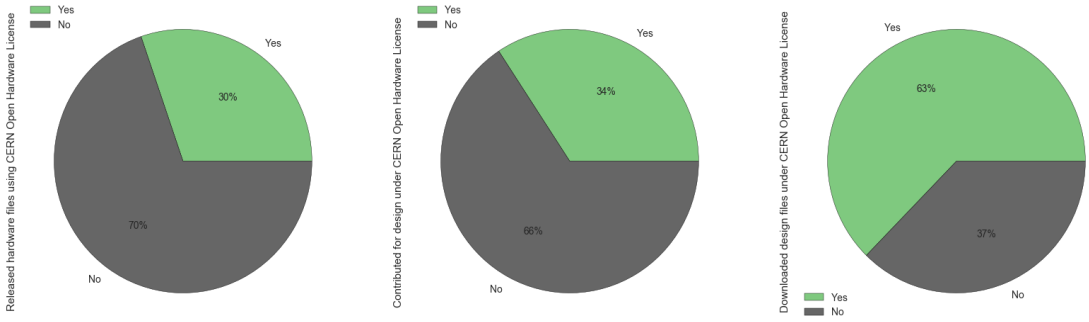
```
In [103]: perceptions3 = dt.ix[:,39:42]
          results = perceptions3.apply(pd.value_counts).fillna(0).T
          fig = results.plot(kind='barh', stacked=True, colormap='PuBu', alpha=0.5,
                             xlim=(0,150))
          fig.legend(('Strongly Disagree','Disagree', 'Mildly Disagree', 'Neutral',
                      'Mildly Agree', 'Agree', 'Strongly Agree'), loc='lower right'

          fig.figure.tight_layout()
          fig.figure.savefig('/tmp/comm_perceptions3.png', dpi=300)
```



1.2.13 Experiences with CERN OHL

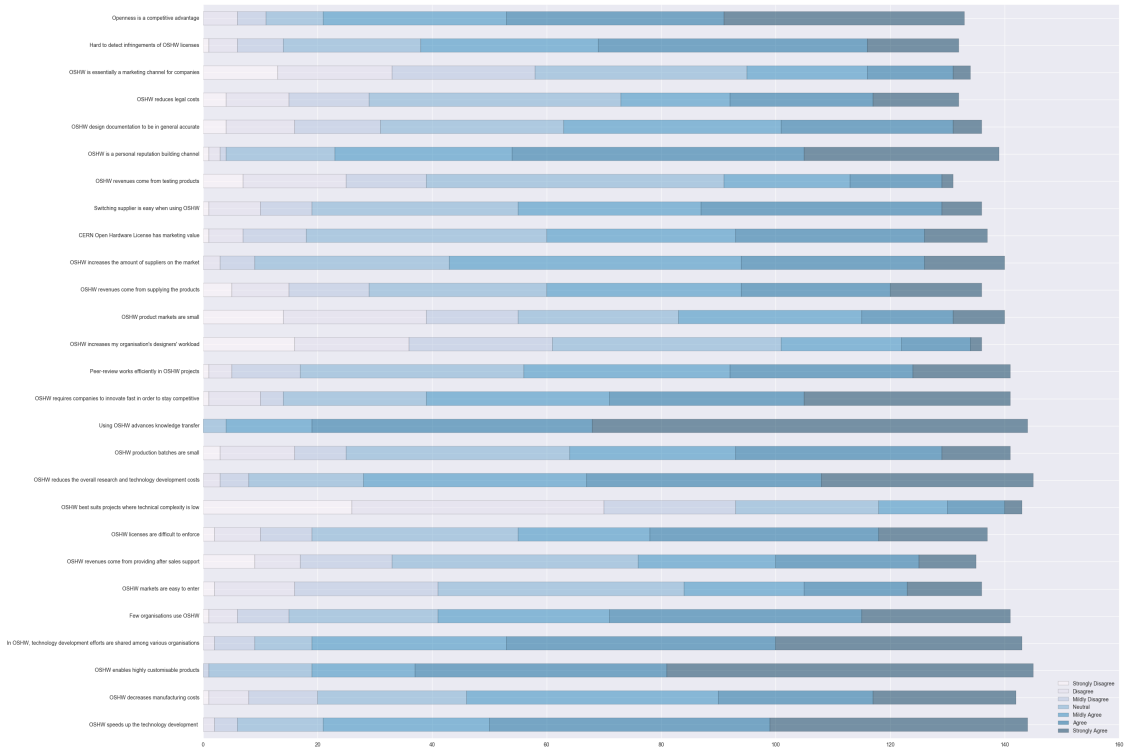
```
In [15]: experiences_ohl = dt.ix[:,26:29]
         ohl_labels = ['Yes', 'No']
```

```
results = experiences_ohl.apply(pd.value_counts)
fig = results.plot(kind='pie', subplots=True, labels=ohl_labels, figsize=(
                   autopct='%.0f%%', colormap='Accent')
#fig.savefig('/tmp/experiences_ohl.png', dpi=300)

# Bar graph
#results = experiences_ohl.apply(pd.value_counts).T
#fig = results.plot(kind='barh', stacked=False, colormap='Accent', figsize
#fig.legend(('Yes', 'No'), loc='upper right')
```



1.2.14 Community Perception (all questions included, extra)

```
In [238]: percept1 = dt.ix[:, 9:26].apply(pd.value_counts).fillna(0)
          percept2 = dt.ix[:, 29:36].apply(pd.value_counts).fillna(0)
          percept3 = dt.ix[:, 39:42].apply(pd.value_counts).fillna(0)
          percept_all = pd.concat([percept1, percept2, percept3], axis=1).T
          fig = percept_all.plot(kind='barh', stacked=True, alpha=0.5, colormap='Pu
                                 xlim=(0,160))
          fig.legend(('Strongly Disagree','Disagree', 'Mildly Disagree', 'Neutral',
                      'Mildly Agree', 'Agree', 'Strongly Agree'), loc='lower right'
          fig.figure.tight_layout()
          fig.figure.savefig('/tmp/comm_perception_all.png', dpi=300)
```



1.2.15 Qualitative Analysis: Code Co-occurrence

```
In [135]: mat = qualicodes.fillna('0')
          #mat
```

2 License

This notebook has been prepared by Luis Felipe R. Murillo [lfmurillo@cnam.fr] for the CERN-KT research collaboration. The survey design and the data collection was conducted by Pietari Kauttu [pietari.kauttu@cern.ch] from the CERN-KT group. This document is licensed as CC-BY-SA 4.0-international, 2017.