Minicourse Analysis

Unknown Author

October 16, 2014

In [127]:

```
from  pymongo import MongoClient
from pylab import *
%matplotlib inline
```

In [128]:

```
client = MongoClient()
db = client.ouroboros
```

In [129]:

```
planet_hunter_proj_id = "5333063b3ae740228a000001"
planet_hunter_users = db.planet_hunter_users
```

1 Basic stats

These are some basic stats about how many users opted in and out of the minicoruse

In [130]:

```
total_users = planet_hunter_users.count()
no_opt_in  = planet_hunter_users.find({"preferences.planet_hunter.course" : "yes"}).co
no_opt_out = planet_hunter_users.find({"preferences.planet_hunter.course" : "no"}).cou

print str(no_opt_in) +", "+ str(no_opt_in*100.0/total_users) + "% users opted in"
print str(no_opt_out) +", "+ str(no_opt_out*100.0/total_users) +"% users opted out"
```

```
1043, 47.0667870036% users opted in
1160, 52.3465703971% users opted out
```

2 Course progress

These plots show for the users who took the course how much they participated. So for example about 65% of users who opted in to the course viewed only 1 slide with only roughtly 5% completing the coruse

In [131]:

```
course_progress   = [int(ph["preferences"]["planet_hunter"]["curr_course_id"]) for ph i
```

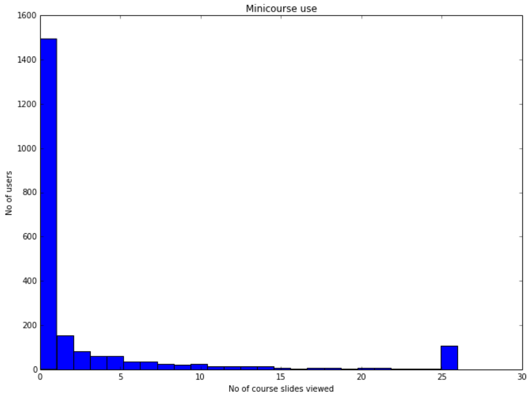
In [132]:

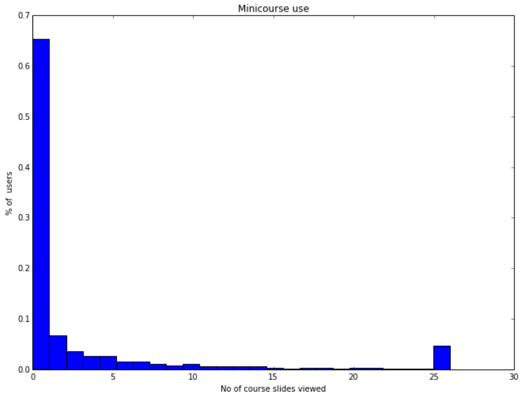
```
fig, axes = plt.subplots(1, 2, figsize=(24,8))

axes[0].set_title("Minicourse use")
axes[0].set_xlabel("No of course slides viewed")
axes[0].set_ylabel("No of users")
axes[0].hist(course_progress,bins=25)


axes[1].set_title("Minicourse use")
axes[1].set_xlabel("No of course slides viewed")
axes[1].set_ylabel("% of  users")
axes[1].hist(course_progress, normed=True, bins=25)
```

```
          (array([ 0.65267536,  0.06727711,  0.03538601,  0.02708559,
Out [132]:0.02621186,
                  0.01485339,  0.01529025,  0.01135847,  0.00830042,
          0.01092161,
                  0.00567924,  0.00524237,  0.0061161 ,  0.00655297,
          0.00349491,
                  0.00131059,  0.00349491,  0.00305805,  0.00087373,
          0.00262119,
                  0.00262119,  0.00174746,  0.00131059,  0.00174746,
          0.04630762]),
           array([  0.  ,   1.04,   2.08,   3.12,   4.16,   5.2 ,   6.24,
          7.28,
                   8.32,   9.36,  10.4 ,  11.44,  12.48,  13.52,  14.56,  15.6 ,
                  16.64,  17.68,  18.72,  19.76,  20.8 ,  21.84,  22.88,  23.92,
                  24.96,  26.  ]),
           <a list of 25 Patch objects>)
```



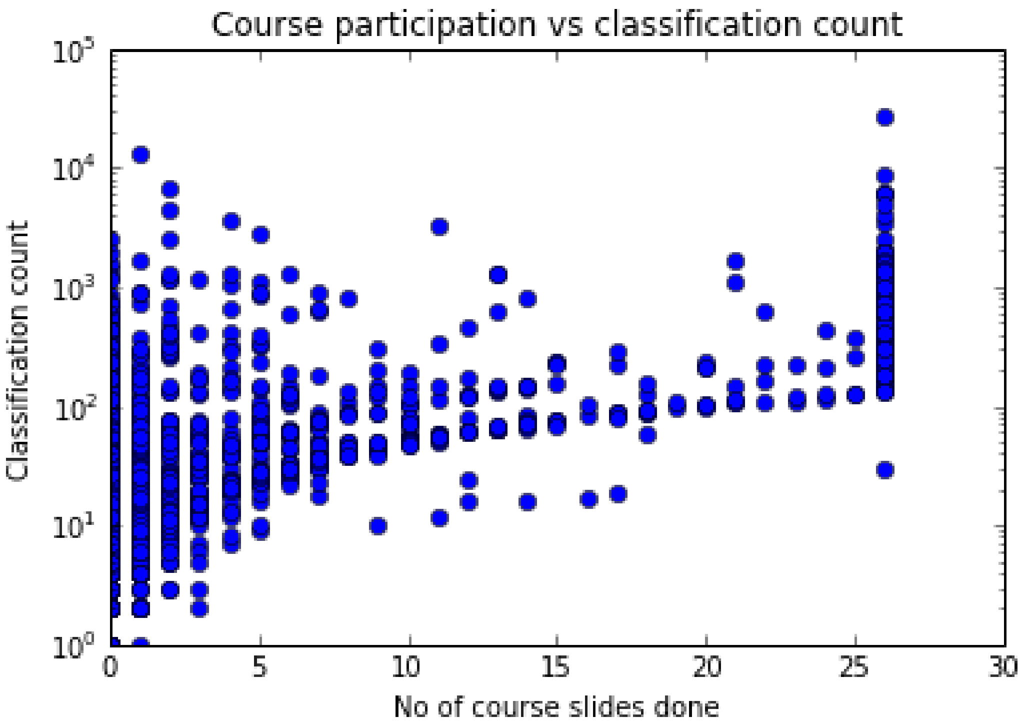


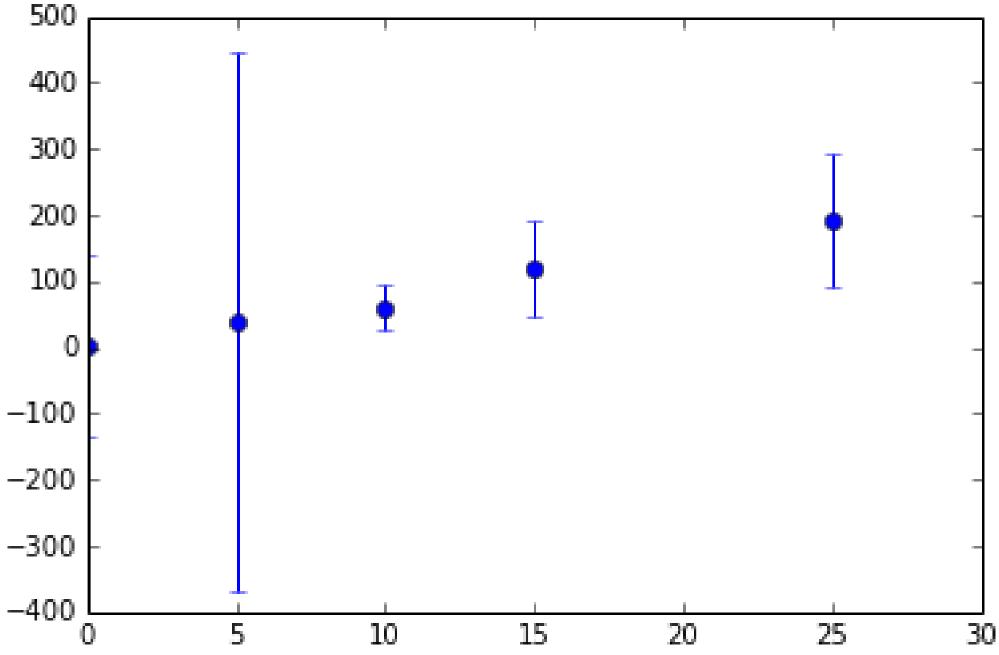
```
In [152]: fig = plt.figure()
          ax = plt.gca()
          ax.set_yscale('log')
          ax.set_title("Course participation vs classification count")
          ax.set_xlabel("No of course slides done")
          ax.set_ylabel("Classification count")

          classification_counts   = [int(ph["projects"][planet_hunter_proj_id].get("classificati
                                        ) for ph in planet_hunter_users.find({"preferences.plan
          ax.plot(course_progress, classification_counts, 'o')

          mean_fig = plt.figure()
          mean_fig_ax = mean_fig.gca()
          together = zip(course_progress, classification_counts)
          means = [median([c_count for course_no,c_count in together if course_no == i]) for i i
          stds  = [std([c_count for course_no,c_count in together if course_no == i]) for i in [

          mean_fig_ax.errorbar([0,5,10,15,25], means, fmt="o", yerr=stds)
          mean_fig_ax.set_yscale('
          <Container object of 3 artists>
Out [152]:
```





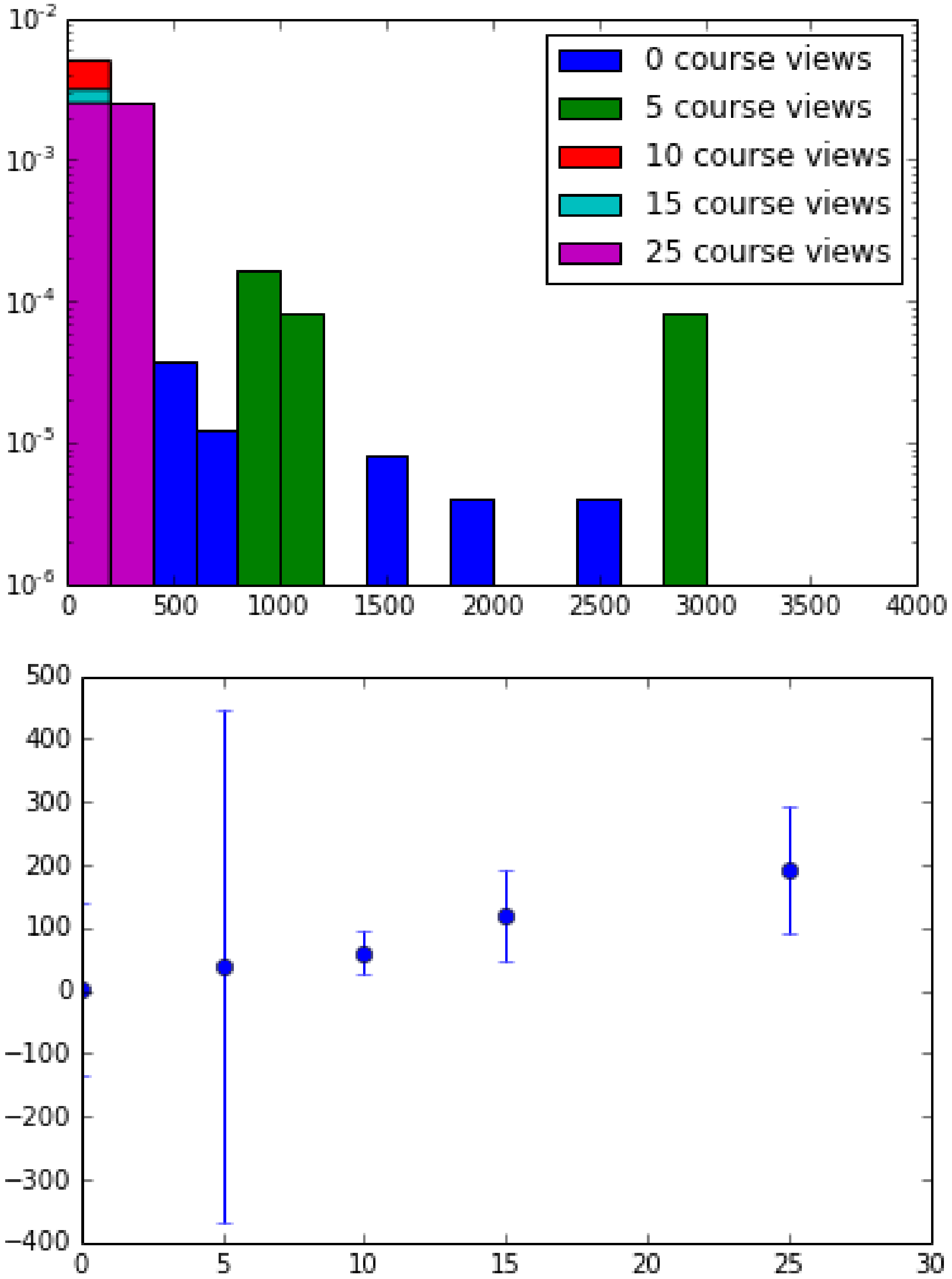
In [150]:
```
hist_fig = plt.figure()
hist_fig_ax = hist_fig.gca()




for i in [0,5,10,15,25]:
    hist_fig_ax.hist( [c_count for course_no,c_count in together if course_no == i], n
hist_fig_ax.legend()
```

```
<Container object of 3 artists>
```

Out [150]:



3 Splits

```
In [ ]:
```