In [2]:

```
%matplotlib inline
from IPython.display import HTML, display
from cStringIO import StringIO
from IPython.display import Javascript
import seaborn as sns

display(HTML("<style>.container { width:100% !important; }</style>"))

#require.config({
#  paths: {
#      d3: r'http://cdnjs.cloudflare.com/ajax/libs/d3/3.4.8/d3.min.js',
#      dimple: r'http://cdnjs.cloudflare.com/ajax/libs/dimple/2.2.0/dimpl
e.latest.min.js'
#  }
#});

Javascript("""$.getScript('http://cdnjs.cloudflare.com/ajax/libs/d3/3.4.8/
d3.min.js')""")
Javascript("""$.getScript('http://cdnjs.cloudflare.com/ajax/libs/dimple/2.
2.0/dimple.latest.min.js')""")
```

Out[2]:

In [53]:

```
import pandas as pd
import pandas.io.data as web
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
import json
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
```

In [4]:

```
with open('coding_challenge_dataset.json', 'r') as f:
    data=f.read().replace('\n', '')
    json_object = json.loads(data)
```

In [5]:

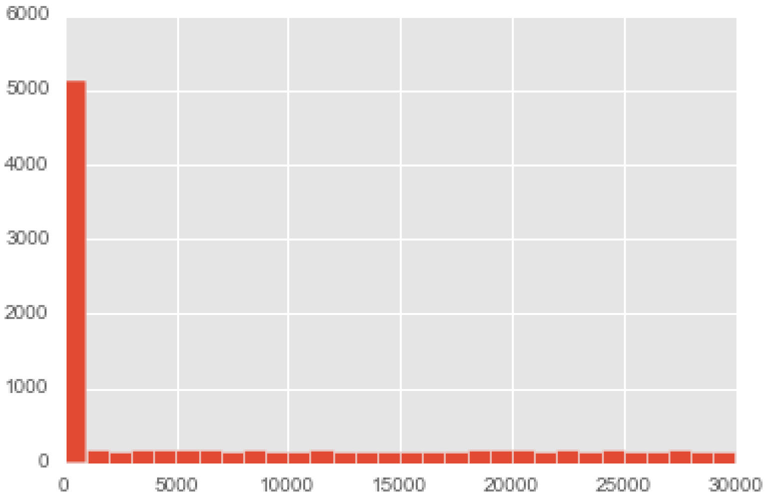
```
data_dict = {
    'serial':[],
    'email':[],
    'livePercent':[],
    'elasticity':[],
    'deadPercent':[],
    'files_input':[],
    'files_output':[],
    'pressure_extruder1':[],
    'pressure_extruder2':[],
    'crosslinking_cl_enabled':[],
    'crosslinking_cl_duration':[],
    'crosslinking_cl_intensity':[],
    'resolution_layerNum':[],
    'resolution_layerHeight':[],
    'wellplate':[]

}
for d in json_object:
    data_dict['serial'].append(d['user_info']['serial'])
    data_dict['email'].append(d['user_info']['email'])
    data_dict['livePercent'].append(d['print_data']['livePercent'])
    data_dict['elasticity'].append(d['print_data']['elasticity'])
    data_dict['deadPercent'].append(d['print_data']['deadPercent'])
    data_dict['files_input'].append(d['print_info']['files']['input'])
    data_dict['files_output'].append(d['print_info']['files']['output'])
    data_dict['pressure_extruder1'].append(d['print_info']['pressure']['extruder1'])
    data_dict['pressure_extruder2'].append(d['print_info']['pressure']['extruder2'])
    data_dict['crosslinking_cl_enabled'].append(d['print_info']['crosslinking']['cl_enabled'])
    data_dict['crosslinking_cl_duration'].append(d['print_info']['crosslinking']['cl_duration'])
    data_dict['crosslinking_cl_intensity'].append(d['print_info']['crosslinking']['cl_intensity'])
    data_dict['resolution_layerNum'].append(d['print_info']['resolution']['layerNum'])
    data_dict['resolution_layerHeight'].append(d['print_info']['resolution']['layerHeight'])
    data_dict['wellplate'].append(d['print_info']['wellplate'])
```

In [6]:
```
df = pd.DataFrame(data_dict)
```

In [52]:
```
lookup_test = df.crosslinking_cl_duration.value_counts(normalize = False)
```

In [13]:
```
display(HTML("<h2>crosslinking_cl_duration</h2>"))
plt.hist(df.crosslinking_cl_duration, bins = 30);
```

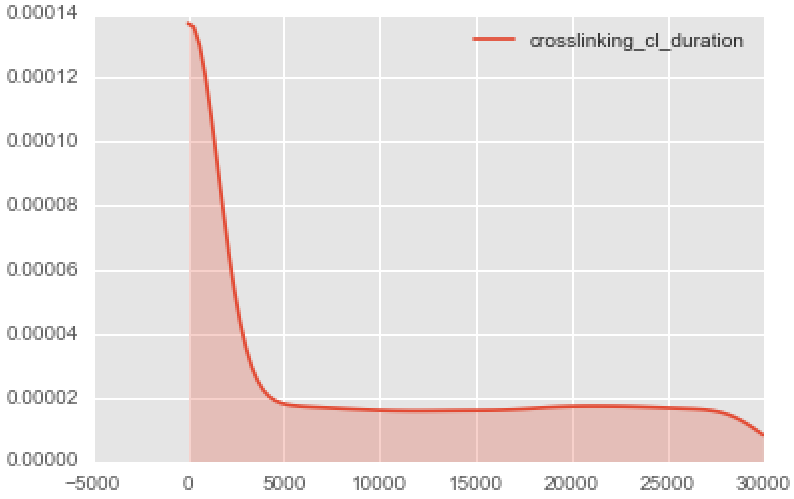
crosslinking_cl_duration



In [48]:
```
len(df.crosslinking_cl_duration.value_counts(normalize = False))
```

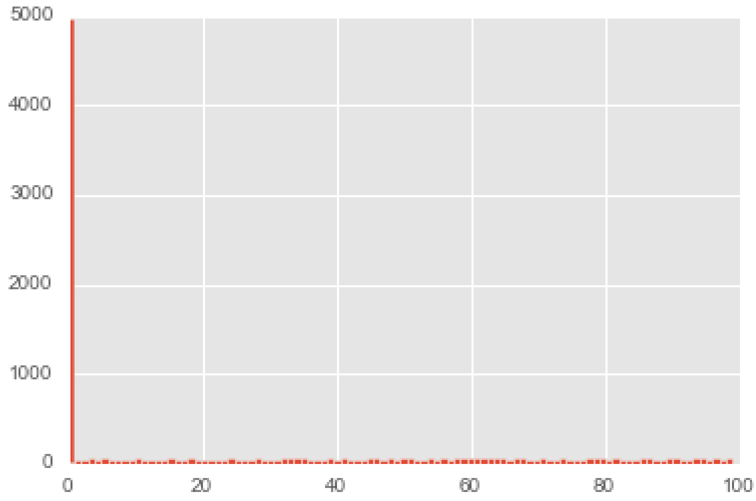
Out[48]:
```
4638
```

In [16]:
```
sns.kdeplot(df.crosslinking_cl_duration, shade=True);
```

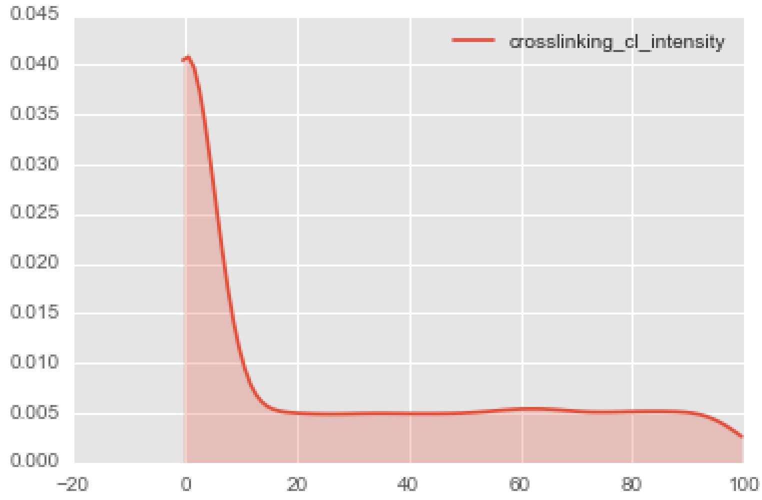


In [44]:
```
display(HTML("<h2>crosslinking_cl_intensity</h2>"))
plt.hist(df.crosslinking_cl_intensity, bins = 100);
#plt.boxplot(df.crosslinking_cl_intensity);
```

crosslinking_cl_intensity



In [20]:
```
sns.kdeplot(df.crosslinking_cl_intensity, shade=True);
```

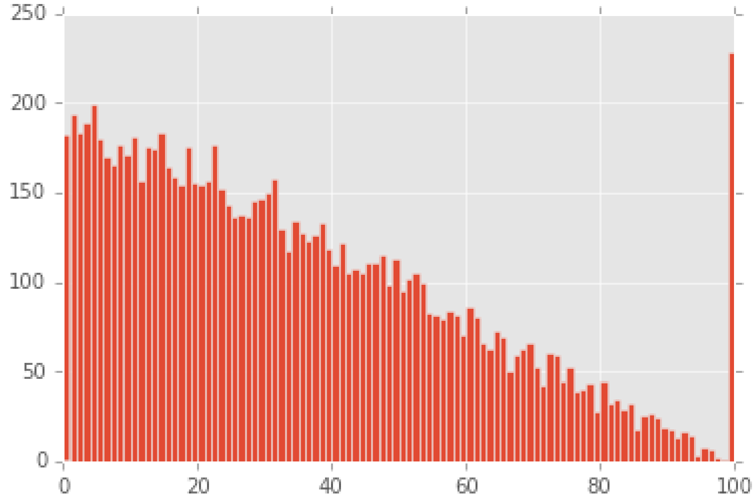


In [47]:
```
len(df.crosslinking_cl_intensity.value_counts(normalize = False))
```

Out[47]:
```
100
```

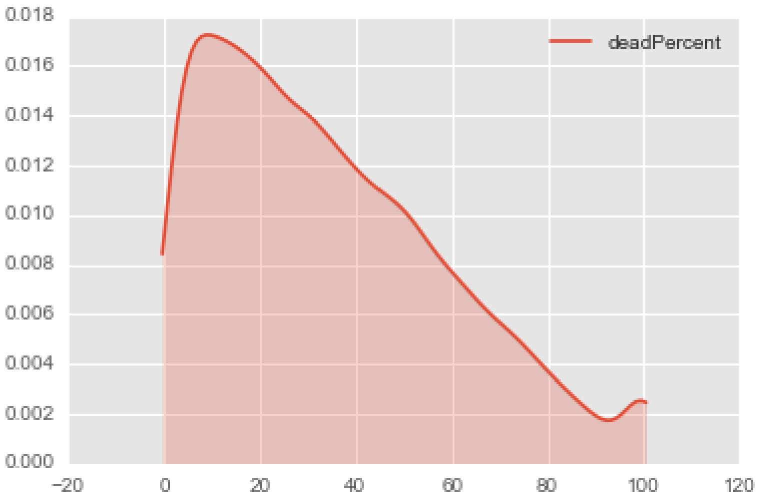
```
In [117]: display(HTML("<h2>deadPercent</h2>"))
          plt.hist(df.deadPercent, bins = 100);
```

deadPercent



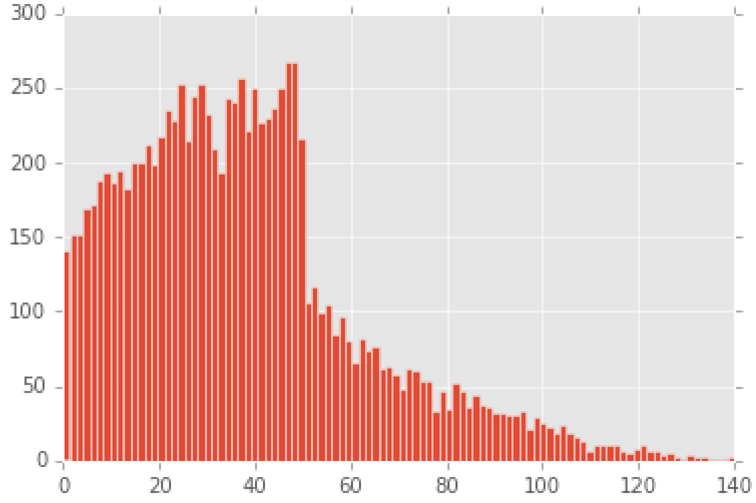
```
In [46]: len(df.deadPercent.value_counts(normalize = False))

Out[46]: 5567
```

```
In [21]: sns.kdeplot(df.deadPercent, shade = True);
```



```
In [118]: display(HTML("<h2>elasticity</h2>"))
          plt.hist(df.elasticity, bins = 100);
```

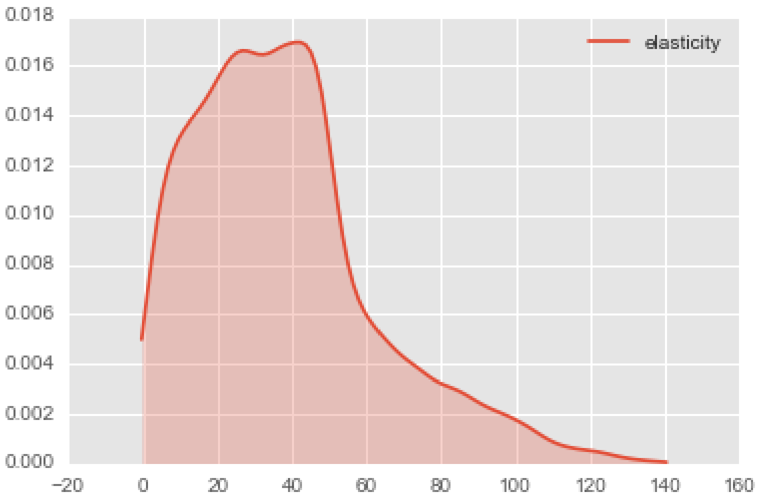
elasticity



```
In [45]: len(df.elasticity.value_counts(normalize = False))
```

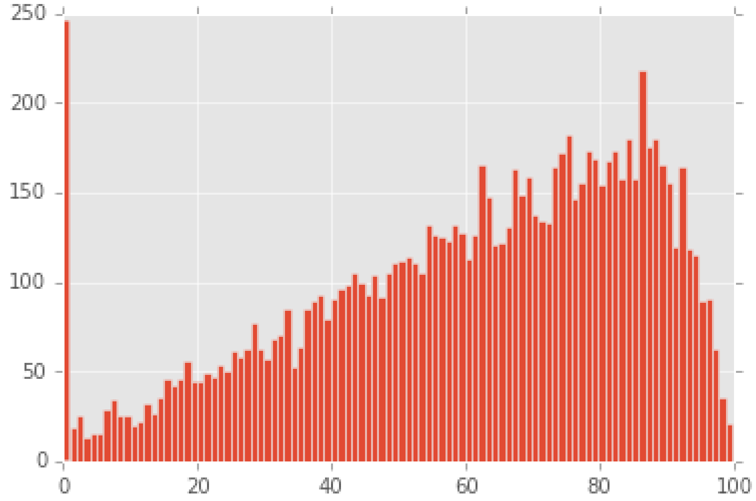
```
Out[45]: 5798
```

```
In [22]: sns.kdeplot(df.elasticity, shade=True);
```



```
In [119]: display(HTML("<h2>livePercent</h2>"))
          plt.hist(df.livePercent, bins = 100);
```

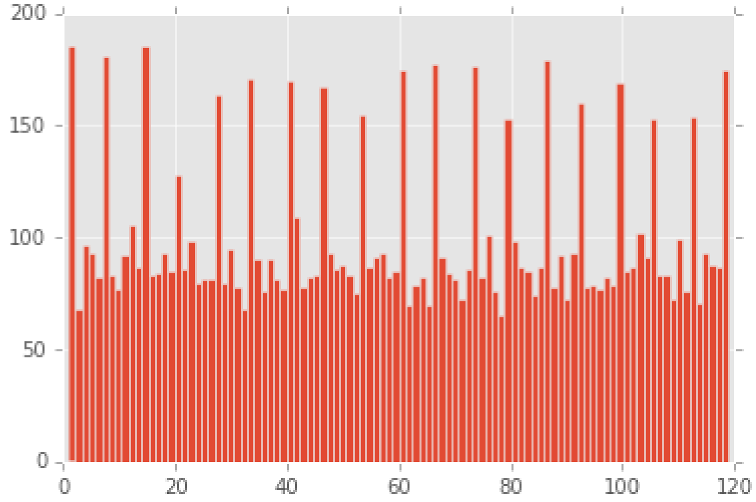
livePercent



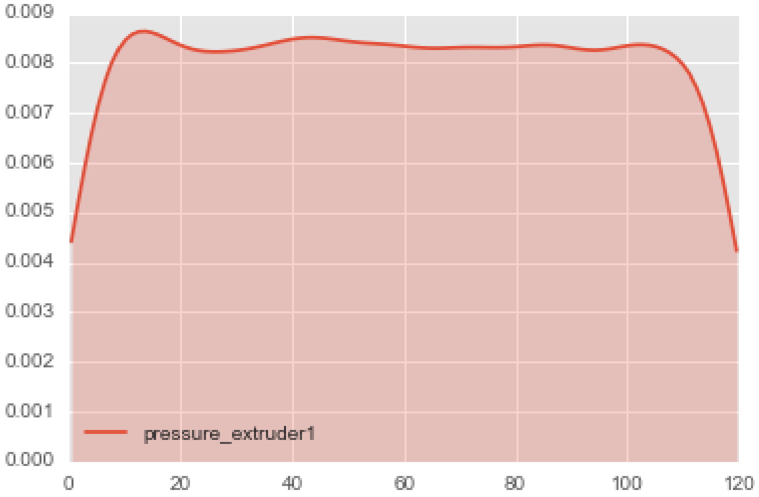
```
In [43]: len(df.livePercent.value_counts(normalize = False))

Out[43]: 5713
```

```
In [120]: display(HTML("<h2>pressure_extruder1</h2>"))
          plt.hist(df.pressure_extruder1, bins = 100);
```

pressure_extruder1



```
In [23]: sns.kdeplot(df.pressure_extruder1, shade=True);
```

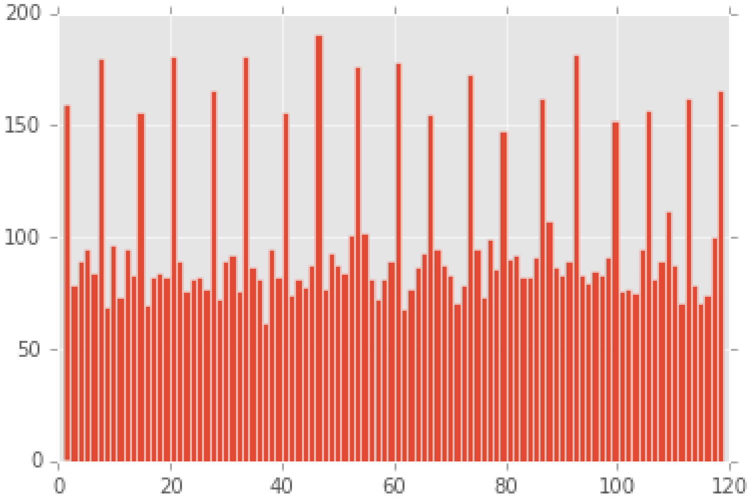


```
In [42]: len(df.pressure_extruder1.value_counts(normalize = True))
```

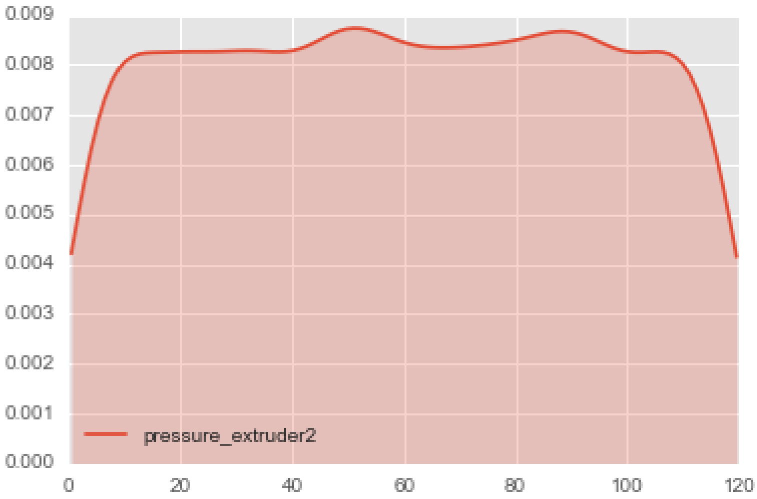
```
Out[42]: 119
```

```
In [121]: display(HTML("<h2>pressure_extruder2</h2>"))
          plt.hist(df.pressure_extruder2, bins = 100);
```

pressure_extruder2



In [24]:
```
sns.kdeplot(df.pressure_extruder2, shade=True);
```

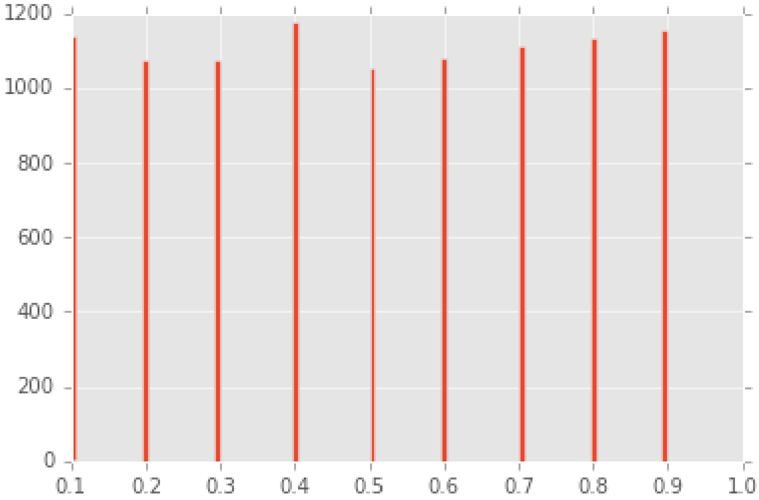


In [41]:
```
len(df.pressure_extruder2.value_counts(normalize = True))
```

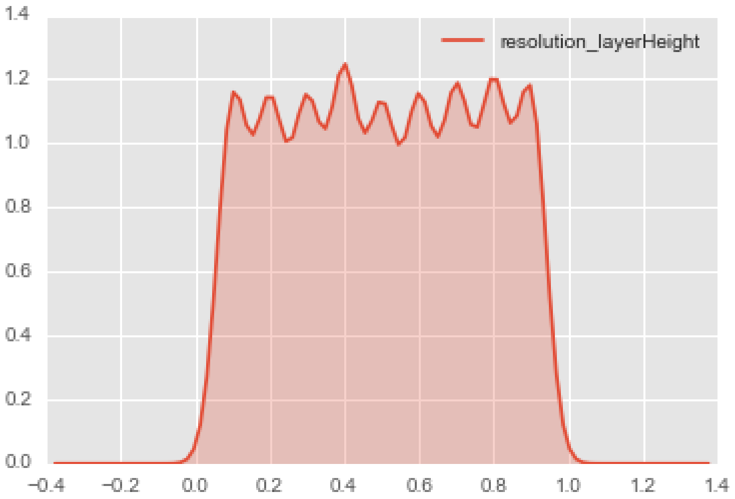
Out[41]:
```
119
```

In [122]:
```
display(HTML("<h2>resolution_layerHeight</h2>"))
plt.hist(df.resolution_layerHeight, bins = 100);
```

resolution_layerHeight



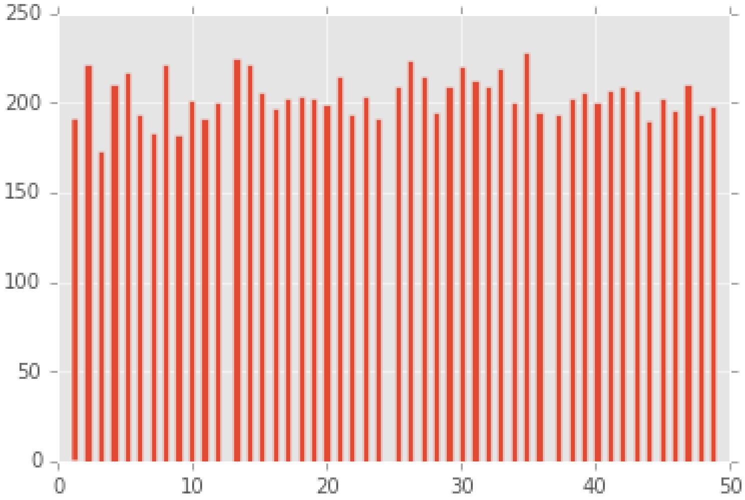
```
In [25]: sns.kdeplot(df.resolution_layerHeight, shade=True);
```



```
In [38]: len(df.resolution_layerHeight.value_counts(normalize = True))

Out[38]: 9
```

```
In [115]: display(HTML("<h2>resolution_layerNum</h2>"))
          plt.hist(df.resolution_layerNum, bins = 100);
```

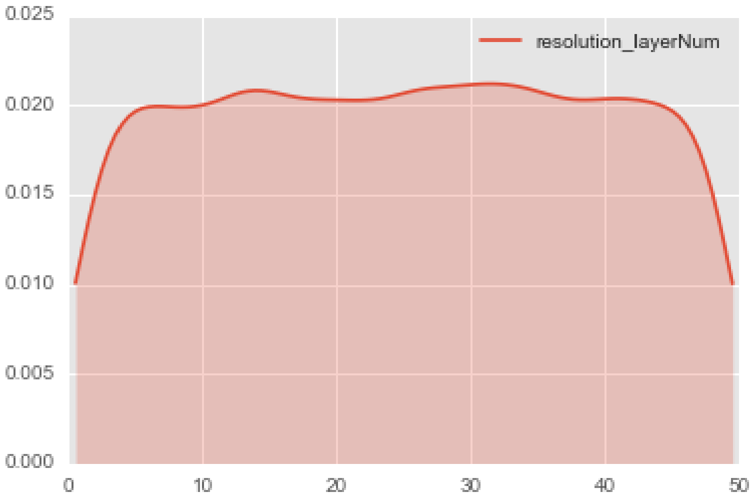
resolution_layerNum



```
In [40]: len(df.resolution_layerNum.value_counts(normalize = True))

Out[40]: 49
```

In [27]:

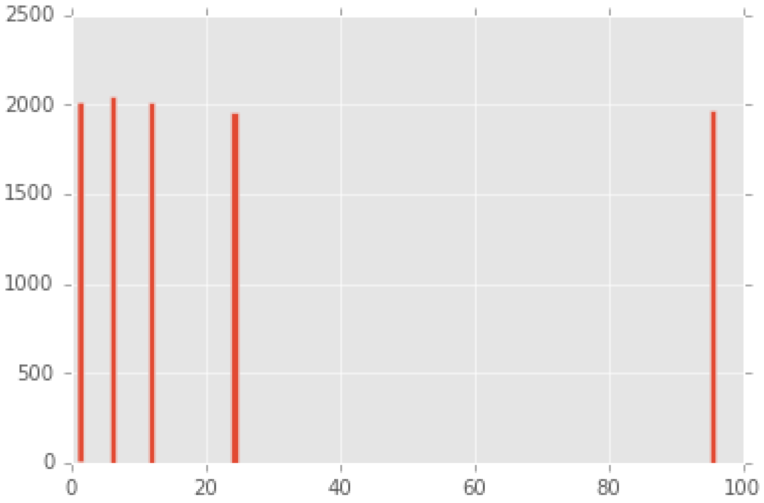
```
sns.kdeplot(df.resolution_layerNum, shade=True);
```



In [123]:

```
display(HTML("<h2>wellplate</h2>"))
plt.hist(df.wellplate, bins = 100);
```

wellplate



In [35]:

```
len(df.wellplate.value_counts(normalize = False))
```

Out[35]:

```
5
```

In [49]:

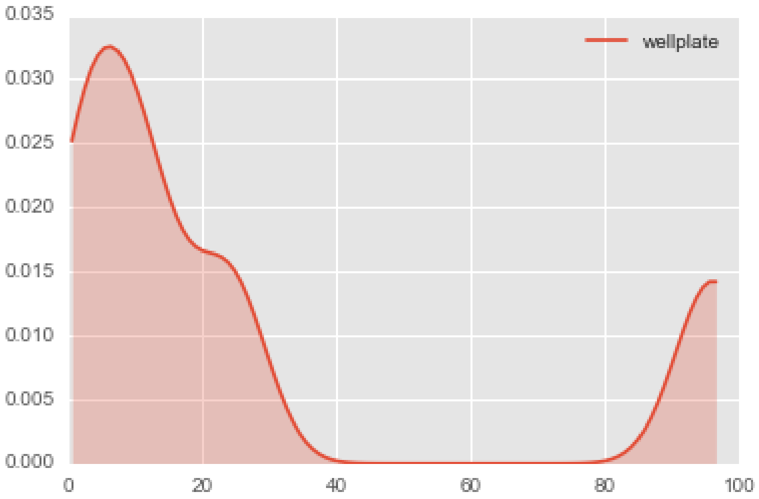
```
df.wellplate.value_counts(normalize = False)
```

Out[49]:

```
6     2044
12    2014
1     2011
96    1972
24    1959
Name: wellplate, dtype: int64
```

In [28]:

```
sns.kdeplot(df.wellplate, shade=True);
```



In [50]:

```
display(HTML("<h2>Quick Stats</h2>"))
df.describe(include = [np.number,'O'])
```

Quick Stats

Out[50]:

	crosslinking_cl_duration	crosslinking_cl_intensity	deadPercent	elasticity
count	10000.00000	10000.000000	10000.000000	10000.00000
unique	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN
mean	7543.55530	25.490500	34.672524	37.678562
std	9713.74483	32.429713	25.010282	24.856149
min	0.00000	0.000000	0.010000	0.020000
25%	0.00000	0.000000	14.010000	19.307500
50%	237.00000	1.000000	30.150000	34.630000
75%	15187.25000	52.000000	51.312500	48.920000
max	29996.00000	99.000000	100.000000	139.900000

In [51]:

```
df.describe(include = [np.number,'O']).to_html(open('my_file.html', 'w'))
```

In [94]:

```
%%javascript
element.append("<div id='chart1'></div>");
```

In [91]:

```
shared_json = df.head(100).to_json()
from IPython.display import HTML
```

In [92]:

```
disp_html = """
<div id="g1" class="container">

</div>
"""

javascript = """
<script type="text/Javascript">

$(document).ready(function(){

var command = "shared_json";
function callback(out_data){
    window.df = out_data;
    var df_local = df;
    df_local = df_local.content.data["text/plain"];
    df_local = df_local.substr(1, df_local.length-2);
    df_local =  $.parseJSON(df_local);
    window.df_json = df_local;

    $('#g1').show();
    $('#g1').css("background-color","#e0ebeb");
    $('#g1').append( "<strong>Hello</strong>" );
}
var callbacks = { 'iopub' : {'output' : callback}};
var kernel = IPython.notebook.kernel;
kernel.execute(command, callbacks, {silent:false});

});


</script>
"""

HTML(disp_html+javascript)
```

Out[92]:

In [97]:

```
javascript = """
<script type="text/Javascript">
$(document).ready(function(){
    $('#g1').css("background-color","#ff6666");
    $('#g1').append( "<strong>"+JSON.stringify(df_json)+"</strong>" );
});
</script>
"""
HTML(javascript)
```

Out[97]: