In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
data=pd.read_csv('U:/Documents/Gunn Notes/Data Analyst Training/CELL/pima.csv',head
er=0, sep=',')
```

In [4]:
```
print (data.columns)
data.head(5)
```

```
Index(['Index', 'pregnant', 'glucose', 'diastolic', 'triceps', 'insulin',
       'bmi', 'diabetes', 'age', 'test'],
      dtype='object')
```

Out[4]:

	Index	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
0	1	6	148	72	35	0	33.6	0.627	50	1
1	2	1	85	66	29	0	26.6	0.351	31	0
2	3	8	183	64	0	0	23.3	0.672	32	1
3	4	1	89	66	23	94	28.1	0.167	21	0
4	5	0	137	40	35	168	43.1	2.288	33	1

QUESTIONS

In [5]:
```
data.describe()
```

Out[5]:

	Index	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.00
mean	384.500000	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.24
std	221.846794	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.76
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.00
25%	192.750000	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.00
50%	384.500000	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.00
75%	576.250000	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.00
max	768.000000	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.00

Question 1: What is the mean insulin amount for patients in the study? Answer: 20.54

Question 2: What was the 3rd quartile of BMI? Answer: 36.60

Question 3: Check out the minimum values for glucose, diastolic BP, triceps, insulin and bmi Answer: Values are 0

In [6]:
```
data.loc[data['glucose'] == 0, 'glucose'] = np.nan
data.loc[data['diastolic'] == 0, 'diastolic'] = np.nan
data.loc[data['triceps'] == 0, 'triceps'] = np.nan
data.loc[data['insulin'] == 0, 'insulin'] = np.nan
data.loc[data['bmi'] == 0, 'bmi'] = np.nan
```

In [8]:
```
print (data.columns)
data.head(10)
```

```
Index(['Index', 'pregnant', 'glucose', 'diastolic', 'triceps', 'insulin',
       'bmi', 'diabetes', 'age', 'test'],
      dtype='object')
```

Out[8]:

	Index	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
0	1	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	2	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	3	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	4	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	5	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
5	6	5	116.0	74.0	NaN	NaN	25.6	0.201	30	0
6	7	3	78.0	50.0	32.0	88.0	31.0	0.248	26	1
7	8	10	115.0	NaN	NaN	NaN	35.3	0.134	29	0
8	9	2	197.0	70.0	45.0	543.0	30.5	0.158	53	1
9	10	8	125.0	96.0	NaN	NaN	NaN	0.232	54	1

In [9]:
```
data.describe()
```

Out[9]:

	Index	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	
count	768.000000	768.000000	763.000000	733.000000	541.000000	394.000000	757.000000	768.000000	768
mean	384.500000	3.845052	121.686763	72.405184	29.153420	155.548223	32.457464	0.471876	33
std	221.846794	3.369578	30.535641	12.382158	10.476982	118.775855	6.924988	0.331329	11
min	1.000000	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21
25%	192.750000	1.000000	99.000000	64.000000	22.000000	76.250000	27.500000	0.243750	24
50%	384.500000	3.000000	117.000000	72.000000	29.000000	125.000000	32.300000	0.372500	29
75%	576.250000	6.000000	141.000000	80.000000	36.000000	190.000000	36.600000	0.626250	41
max	768.000000	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81

Question 4: What is the mean insulin amount for patients in the study now? Answer: 155.54

Question 5: What is the 3rd quartile of BMI now? Answer: 36.60

Question 6: Check out the minimum values for glucose, diastolic BP, triceps, insulin and bmi now Answer: Non zero values are dispalyed now

In [10]:
```
#diabetes 0-no, 1-yes
data['test'] = data['test'].replace(0, 'No')
data['test'] = data['test'].replace(1, 'Yes')
```

In [12]:
```
print (data.columns)
data.head(10)
```

```
Index(['Index', 'pregnant', 'glucose', 'diastolic', 'triceps', 'insulin',
       'bmi', 'diabetes', 'age', 'test'],
      dtype='object')
```

Out[12]:

	Index	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
0	1	6	148.0	72.0	35.0	NaN	33.6	0.627	50	Yes
1	2	1	85.0	66.0	29.0	NaN	26.6	0.351	31	No
2	3	8	183.0	64.0	NaN	NaN	23.3	0.672	32	Yes
3	4	1	89.0	66.0	23.0	94.0	28.1	0.167	21	No
4	5	0	137.0	40.0	35.0	168.0	43.1	2.288	33	Yes
5	6	5	116.0	74.0	NaN	NaN	25.6	0.201	30	No
6	7	3	78.0	50.0	32.0	88.0	31.0	0.248	26	Yes
7	8	10	115.0	NaN	NaN	NaN	35.3	0.134	29	No
8	9	2	197.0	70.0	45.0	543.0	30.5	0.158	53	Yes
9	10	8	125.0	96.0	NaN	NaN	NaN	0.232	54	Yes

In [13]:
```
data.dtypes
```

Out[13]:
```
Index          int64
pregnant       int64
glucose      float64
diastolic    float64
triceps      float64
insulin      float64
bmi          float64
diabetes     float64
age            int64
test          object
dtype: object
```

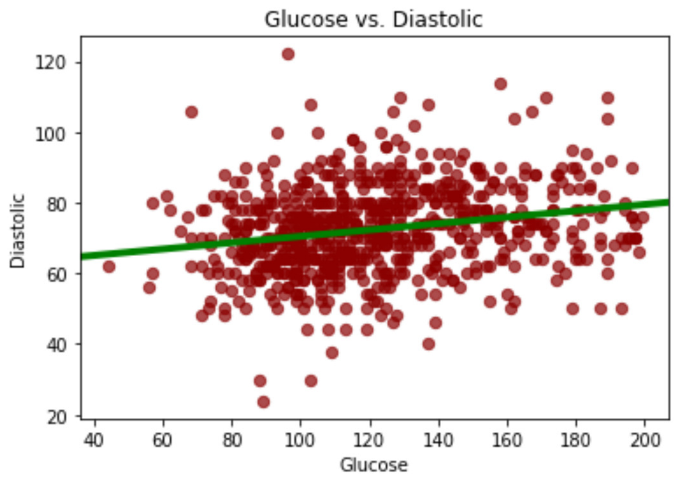
In [14]:
```
#df['col_name'] = df['col_name'].astype('category')
data.test.value_counts()
```

Out[14]:
```
No     500
Yes    268
Name: test, dtype: int64
```

Question 7: use the table function on the test column to determine how many in this dataset have diabetes? Answer: 268 patients have diabetes

Plot glucose against diastolic blood pressure (use the plot function)

In [18]:
```
import seaborn as sns
plot = sns.regplot(x="glucose", y="diastolic", data=da
ta, ci = False,
    scatter_kws={"color":"darkred","alpha":0.7,"s":4
0},
    line_kws={"color":"g","alpha":1,"lw":4})
plot.set(xlabel="Glucose", ylabel="Diastolic", title='
Glucose vs. Diastolic');
```



Question 8: Do we get normal (or near normal) distributions? Answer: We get normal distribution

In [26]:
```
g = sns.jointplot(x='glucose', y='diastolic', data=data, kind="reg")
regline = g.ax_joint.get_lines()[0]
regline.set_color('red')
regline.set_zorder('5')
```

```
Error in callback <function install_repl_displayhook.<locals>.post_execute at 0x00000
00009A9D0D8> (for post_execute):
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
P:\Anaconda\lib\site-packages\matplotlib\pyplot.py in post_execute()
    107             def post_execute():
    108                 if matplotlib.is_interactive():
--> 109                     draw_all()
    110 
    111             # IPython >= 2

P:\Anaconda\lib\site-packages\matplotlib\_pylab_helpers.py in draw_all(cls, force)
    126         for f_mgr in cls.get_all_fig_managers():
    127             if force or f_mgr.canvas.figure.stale:
--> 128                 f_mgr.canvas.draw_idle()
    129 
    130 atexit.register(Gcf.destroy_all)

P:\Anaconda\lib\site-packages\matplotlib\backend_bases.py in draw_idle(self, *args,
**kwargs)
   1905         if not self._is_idle_drawing:
   1906             with self._idle_draw_cntx():
-> 1907                 self.draw(*args, **kwargs)
   1908 
   1909     def draw_cursor(self, event):

P:\Anaconda\lib\site-packages\matplotlib\backends\backend_agg.py in draw(self)
    386         self.renderer = self.get_renderer(cleared=True)
    387         with RendererAgg.lock:
--> 388             self.figure.draw(self.renderer)
    389             # A GUI class may be need to update a window using this draw, so
    390             # don't forget to call the superclass.

P:\Anaconda\lib\site-packages\matplotlib\artist.py in draw_wrapper(artist, renderer,
*args, **kwargs)
     36                 renderer.start_filter()
     37 
---> 38             return draw(artist, renderer, *args, **kwargs)
     39         finally:
     40             if artist.get_agg_filter() is not None:

P:\Anaconda\lib\site-packages\matplotlib\figure.py in draw(self, renderer)
   1707             self.patch.draw(renderer)
   1708             mimage._draw_list_compositing_images(
-> 1709                 renderer, self, artists, self.suppressComposite)
   1710 
   1711             renderer.close_group('figure')

P:\Anaconda\lib\site-packages\matplotlib\image.py in _draw_list_compositing_images(re
nderer, parent, artists, suppress_composite)
    133     if not_composite or not has_images:
    134         for a in artists:
--> 135             a.draw(renderer)
    136     else:
    137         # Composite any adjacent images together

P:\Anaconda\lib\site-packages\matplotlib\artist.py in draw_wrapper(artist, renderer,
*args, **kwargs)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
P:\Anaconda\lib\site-packages\IPython\core\formatters.py in __call__(self, obj)
    339                 pass
    340             else:
--> 341                 return printer(obj)
    342             # Finally look for special method names
    343             method = get_real_method(obj, self.print_method)

P:\Anaconda\lib\site-packages\IPython\core\pylabtools.py in <lambda>(fig)
    242
    243     if 'png' in formats:
--> 244         png_formatter.for_type(Figure, lambda fig: print_figure(fig, 'png',
**kwargs))
    245     if 'retina' in formats or 'png2x' in formats:
    246         png_formatter.for_type(Figure, lambda fig: retina_figure(fig, **kwarg
s))

P:\Anaconda\lib\site-packages\IPython\core\pylabtools.py in print_figure(fig, fmt, bb
ox_inches, **kwargs)
    126
    127     bytes_io = BytesIO()
--> 128     fig.canvas.print_figure(bytes_io, **kw)
    129     data = bytes_io.getvalue()
    130     if fmt == 'svg':

P:\Anaconda\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filen
ame, dpi, facecolor, edgecolor, orientation, format, bbox_inches, **kwargs)
   2054                         orientation=orientation,
   2055                         dryrun=True,
-> 2056                         **kwargs)
   2057                     renderer = self.figure._cachedRenderer
   2058                     bbox_artists = kwargs.pop("bbox_extra_artists", None)

P:\Anaconda\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, f
ilename_or_obj, metadata, pil_kwargs, *args, **kwargs)
    525
    526         else:
--> 527             FigureCanvasAgg.draw(self)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \

P:\Anaconda\lib\site-packages\matplotlib\backends\backend_agg.py in draw(self)
    386         self.renderer = self.get_renderer(cleared=True)
    387         with RendererAgg.lock:
--> 388             self.figure.draw(self.renderer)
    389             # A GUI class may be need to update a window using this draw, so
    390             # don't forget to call the superclass.

P:\Anaconda\lib\site-packages\matplotlib\artist.py in draw_wrapper(artist, renderer,
*args, **kwargs)
     36                 renderer.start_filter()
     37
---> 38             return draw(artist, renderer, *args, **kwargs)
     39         finally:
     40             if artist.get_agg_filter() is not None:
```

```
<Figure size 432x432 with 3 Axes>
```

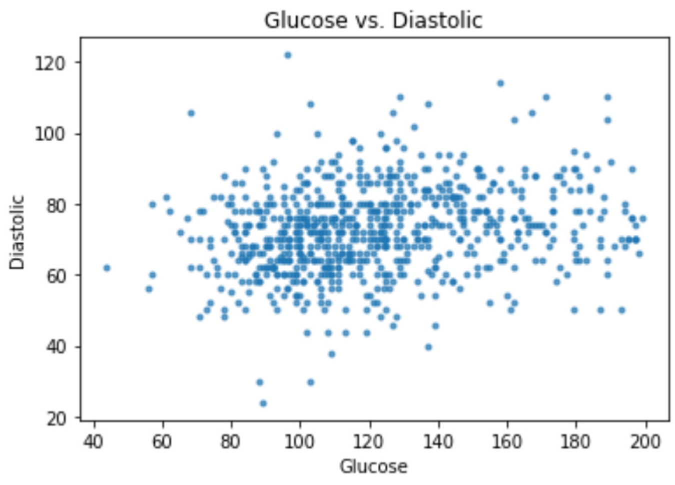
In [43]:

```
from matplotlib import pyplot as plt
#scatterplot of glucose against diastolic
x = data.glucose
y = data.diastolic
colors = ('#d62728', '#9467bd')
# '#d62728', '#9467bd'
#'red', 'green'
area = np.pi*3

# Plot
plt.scatter(x, y,s=area, alpha=0.7)
#c=colors,
plt.title('Glucose vs. Diastolic')
plt.xlabel('Glucose')
plt.ylabel('Diastolic')

plt.show()

print("Glucose=red and Diastolic=green")
```



```
Glucose=red and Diastolic=green
```

In [24]:

```
import numpy as np
#masked arrey numpy module
#np.corrcoef not NaN tolerant
import numpy.ma as ma
print(ma.corrcoef(ma.masked_invalid(data.glucose), ma.masked_invalid(data.diastolic)))
```

```
[[1.0 0.22319177824954192]
 [0.22319177824954192 1.0]]
```

In [ ]: