Case Study 1 Questions

What is the difference in weight between babies born to mothers who smoked during pregnancy and those who did not? Is this difference important to the health of the baby?

1. Summarize numerically the two distributions of birth weight for babies born to women who smoked during their pregnancy and for babies born to women who did not smoke during their pregnancy.
2. Use graphical methods to compare the two distributions of birth weight.
3. Compare the frequency, or incidence, of low-birth-weight babies for the two groups. How reliable do you think your estimates are? That is, how would the incidence of low birth weight change if a few more or fewer babies were classified as low birth weight?
4. Asses the importance of the difference you found in your three types of comparisons (numerical, graphical, incidence). Summarize your findings and relate them to other studies.
5. Form own question for last part.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from scipy.stats import ttest_ind
from scipy.stats.mstats import mquantiles
from scipy.stats import kurtosis
from scipy.stats import probplot
from scipy.stats import ttest_ind
from scipy.stats import *
```

In [2]:
```
# Set upt plot style
plt.style.use(['fivethirtyeight'])
```

Load datasets

In [3]:
```
print(plt.style.available)
```

```
['fivethirtyeight', 'seaborn-colorblind', 'dark_background', 'seaborn-dar
k', 'seaborn-bright', 'seaborn-poster', 'seaborn-talk', 'bmh', 'seaborn-w
hitegrid', 'seaborn-ticks', 'ggplot', 'seaborn-white', 'seaborn-paper',
'seaborn-dark-palette', 'seaborn-pastel', 'seaborn-muted', 'seaborn', 'se
aborn-deep', 'seaborn-darkgrid', 'seaborn-notebook', 'grayscale', '_class
ic_test', 'classic']
```

In [4]:
```
# Load in the data sets
df1 = pd.read_csv('babies.txt', delim_whitespace=True)
df2 = pd.read_csv('babies23.txt', delim_whitespace=True)
```

In [5]:
```
df1 = df1[df1.weight != 999]
```

Exploring 'babies.txt'

bwt - birth weight in ounces (999 unknown)

gestation - gestation days

parity - 0 means first born

age - mom age in years

height - mom height in inches

weight - mom pre-pregnancy weight in pounds

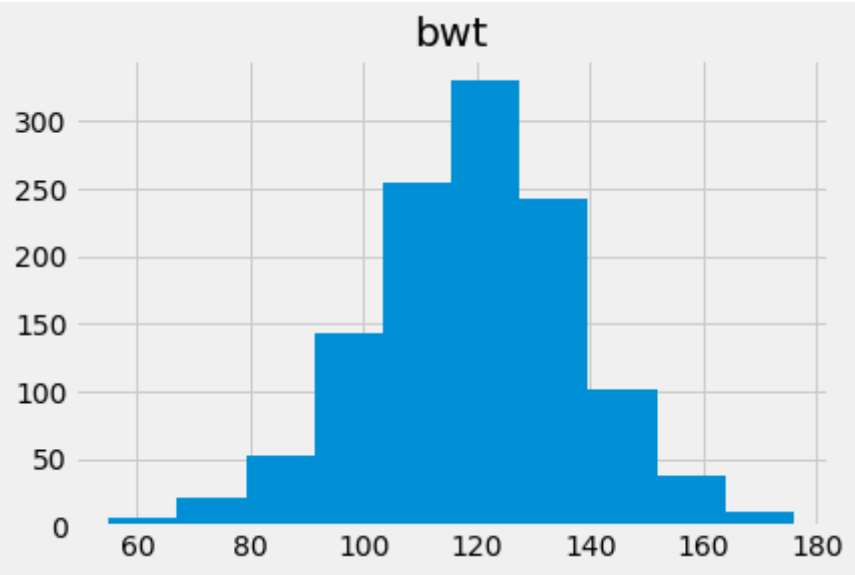
smoke - mom smoke, 0 means no, 1 means yes, 9 means unknown

In [6]:
```
df1.head()
```

Out[6]:

	bwt	gestation	parity	age	height	weight	smoke
0	120	284	0	27	62	100	0
1	113	282	0	33	64	135	0
2	128	279	0	28	64	115	1
3	123	999	0	36	69	190	0
4	108	282	0	23	67	125	1

In [7]:
```
for c in df1.columns:
    df1.hist(column=c)[0]
    plt.show()
```



Exploring 'babies23.txt'

id - id number

pluralty - 5 means single fetus

outcome - 1 for live birth that survived at least 28 days

date - birth date 1096=January 1, 1961 (this might be a timestamp, not very sure)

gestation - gestation days

sex - infant sex, 1=male, 2=female, 9=unknown

wt - birth weight in ounces

parity - 0 means first born

race - mom race, 0-5=white, 6=mex, 7=black, 8=asian, 9=mix, 99=unknown

age - mom age in years

ed - mom education, 0=(<8), 1=(8-<12), 2=12, 3=12+trade, 4=12+some college, 5=16, 7=trade (hs unclear), 9=unknown

ht - mom height in inches

wt - mom pre-pregnancy weight in pounds (notice that this column name will be renamed to wt.1, since there are two duplicate wt column names)

drace - dad race

dage - dad age

ded - dad education

dht - dad height

dwt - dad weight

marital - 1=married, 2-4=sep, div, wid, 5=never married, blank

inc - total income in 2500 increments, 0=under 2500, 1=2500-4999, ..., 9=22500+, 98=unknown, 99=not asked

smoke - mom smoke, 0=never, 1=yes now, 2=until pregnancy, 3=once did not now, 9=unknown

time - how long ago quit, 0=never, 1=still, 2=during preg, 3=up to 1 yr, 4=up to 2 yr, 5=up to 3 yr, 6=up to 4 yr, 7=5 to 9 yr, 8=10+ yr, 9=quit and don't know, 98=unknown

number - number of cigs smoke a day for past and current smokers, 0=never, 1=1-4, 2=5-9, 3=10-14, 4=15-19, 5=20-29, 6=30-39, 7=40-60, 8=60+, 9=smoke but don't know, 98=unknown

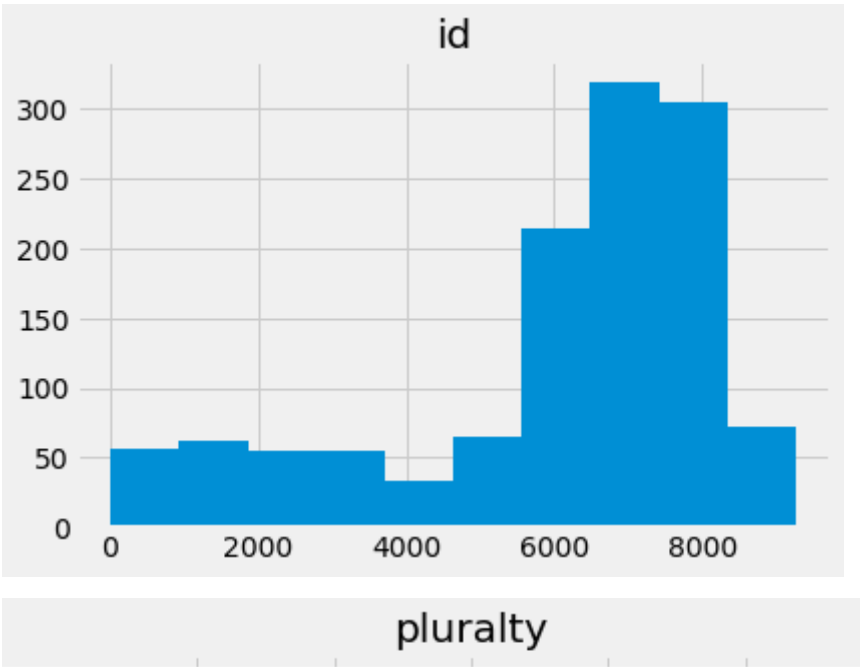
In [8]:
```
df2.head()
```

Out[8]:

	id	pluralty	outcome	date	gestation	sex	wt	parity	race	age	...	drace	dage	ded	dht
0	15	5	1	1411	284	1	120	1	8	27	...	8	31	5	65
1	20	5	1	1499	282	1	113	2	0	33	...	0	38	5	70
2	58	5	1	1576	279	1	128	1	0	28	...	5	32	1	99
3	61	5	1	1504	999	1	123	2	0	36	...	3	43	4	68
4	72	5	1	1425	282	1	108	1	0	23	...	0	24	5	99

5 rows × 23 columns

In [9]:
```
for c in df2.columns:
    df2.hist(column=c)[0]
    plt.show()
```



Question 1

Summarize numerically the two distributions of birth weight for babies born to women who smoked during their pregnancy and for babies born to women who did not smoke during their pregnancy.

What we'll use:

1. Min/max bwt of smoke/nonsmoke
2. mean bwt of smoke/nonesmoke
3. median bwt of smoke/nonsmoke
4. quartiles

Note here we will **ONLY be using 'babies.txt'*

In [10]:
```
df1 = pd.read_csv('babies.txt', delim_whitespace=True)

# Partition data to smoke babies and no smoke babies
smoke_mask = (df1['smoke'] == 1)
no_smoke_mask = (df1['smoke'] == 0)

smoke_df = df1.loc[smoke_mask]
no_smoke_df = df1.loc[no_smoke_mask]
```

In [11]:
```
smoke_df = smoke_df[smoke_df.bwt != 999]
no_smoke_df = no_smoke_df[no_smoke_df.bwt != 999]
smoke_df = smoke_df[smoke_df.gestation != 999]
no_smoke_df = no_smoke_df[no_smoke_df.gestation != 999]
smoke_df = smoke_df[smoke_df.age != 99]
no_smoke_df = no_smoke_df[no_smoke_df.age != 99]
```

In [12]:
```
ttest_ind(smoke_df['bwt'], no_smoke_df['bwt'])
levene(smoke_df['bwt'], no_smoke_df['bwt'])
```

Out[12]:
```
LeveneResult(statistic=3.3643156816753907, pvalue=0.06686890889225966)
```

In [13]:
```
def print_info(cat):
    # Calculate min, max, mean, and median of smoking bwt
    min_smoke_bwt = min(smoke_df[cat])
    max_smoke_bwt = max(smoke_df[cat])
    mean_smoke_bwt = np.mean(smoke_df[cat])
    med_smoke_bwt = np.median(smoke_df[cat])

    # Calculate min, max, mean, and median of non-smoking bwt
    min_no_smoke_bwt = min(no_smoke_df[cat])
    max_no_smoke_bwt = max(no_smoke_df[cat])
    mean_no_smoke_bwt = np.mean(no_smoke_df[cat])
    med_no_smoke_bwt = np.median(no_smoke_df[cat])

    print(cat)
    print('\tsmoke\tno smoke')
    print('min\t%f\t%f' % (min_smoke_bwt,min_no_smoke_bwt))
    print('max\t%f\t%f' % (max_smoke_bwt,max_no_smoke_bwt))
    print('mean\t%f\t%f' % (mean_smoke_bwt,mean_no_smoke_bwt))
    print('med\t%f\t%f' % (med_smoke_bwt, med_no_smoke_bwt))
    print("count\t%d\t%d" % (len(smoke_df[cat]), len(no_smoke_df[cat])))
    print("std\t%.3f\t%.3f" % (smoke_df[cat].std(), no_smoke_df[cat].std()))

    print()
    # Quantiles
    print("smoking %s quantiles" % (cat))
    print(mquantiles(smoke_df[cat].astype('float')))
    print()
    print("non smoking %s quantiles" %(cat))
    print(mquantiles(no_smoke_df[cat].astype('float')))
```

In [14]:
```
print('|---------------bwt--------------------|')
print_info('bwt')
print('|---------------gestation--------------------|')
print_info('gestation')
print('|-----------------age---------------|')
print_info('age')
print('|---------------parity-----------------|')
print_info('parity')
```

```
|---------------bwt--------------------|
bwt
        smoke   no smoke
min     58.000000       55.000000
max     163.000000      176.000000
mean    114.081420      123.154372
med     115.000000      123.500000
count   479     732
std     18.180  17.353

smoking bwt quantiles
[ 101.2  115.   126. ]

non smoking bwt quantiles
[ 113.   123.5  134. ]
|---------------gestation--------------------|
gestation
        smoke   no smoke
min     223.000000      148.000000
max     330.000000      353.000000
mean    277.991649      280.178962
med     279.000000      281.000000
count   479     732
std     15.087  16.638

smoking gestation quantiles
[ 271.  279.  286.]

non smoking gestation quantiles
[ 273.  281.  289.]
|-----------------age---------------|
age
        smoke   no smoke
min     15.000000       17.000000
max     43.000000       45.000000
mean    26.749478       27.536885
med     26.000000       26.500000
count   479     732
std     5.654   5.836

smoking age quantiles
[ 22.  26.  30.]

non smoking age quantiles
[ 23.   26.5  31. ]
|---------------parity-----------------|
parity
        smoke   no smoke
```

```
min     0.000000        0.000000
max     1.000000        1.000000
mean    0.252610        0.260929
med     0.000000        0.000000
count   479     732
std     0.435   0.439

smoking parity quantiles
[ 0.  0.  1.]

non smoking parity quantiles
[ 0.  0.  1.]
```

In [15]:
```
print_info('gestation')
```

```
gestation
        smoke   no smoke
min     223.000000      148.000000
max     330.000000      353.000000
mean    277.991649      280.178962
med     279.000000      281.000000
count   479     732
std     15.087  16.638

smoking gestation quantiles
[ 271.  279.  286.]

non smoking gestation quantiles
[ 273.  281.  289.]
```

Question 2

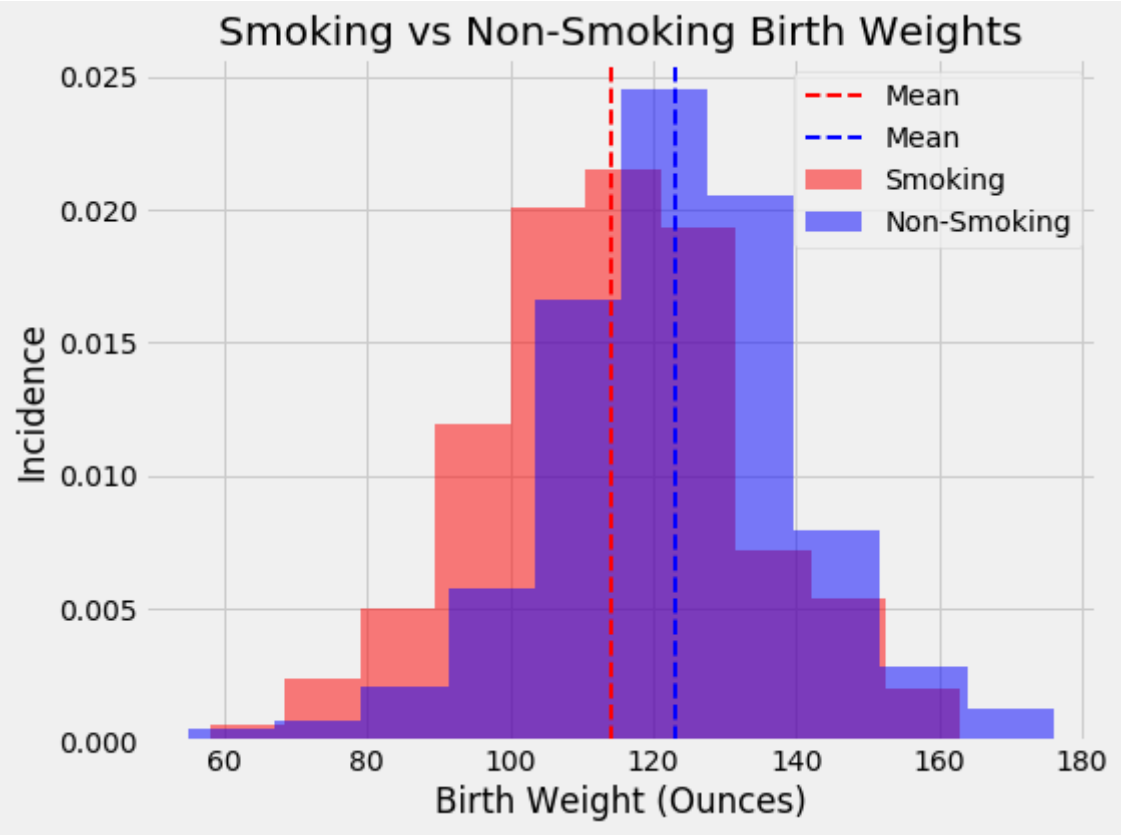
Use graphical methods to compare the two distributions of birth weight.

In [16]:
```
smoke_df['bwt']/max(smoke_df['bwt'])
```

Out[16]:
```
2       0.785276
4       0.662577
9       0.877301
11      0.883436
12      0.865031
13      0.674847
16      0.564417
17      0.705521
19      0.730061
21      0.705521
25      0.631902
27      0.699387
29      0.699387
37      0.822086
38      0.748466
42      0.846626
44      0.533742
45      0.877301
49      0.889571
51      0.662577
56      0.760736
62      0.748466
63      0.619632
64      0.785276
65      0.638037
67      0.840491
68      0.631902
72      0.815951
74      0.558282
76      0.938650
          ...
1175    0.846626
1176    0.539877
1179    0.503067
1182    0.595092
1186    0.711656
1187    0.809816
1189    0.730061
1191    0.650307
1194    0.631902
1195    0.687117
1196    0.588957
1197    0.625767
1198    0.736196
1200    0.595092
1203    0.595092
1206    0.779141
1207    0.533742
1208    0.865031
1210    0.711656
1211    0.460123
1215    0.932515
1218    0.496933
1221    0.699387
1222    0.760736
```

```
1223    0.705521
1224    0.877301
1225    0.693252
1226    0.668712
1227    0.631902
1233    0.797546
Name: bwt, Length: 479, dtype: float64
```

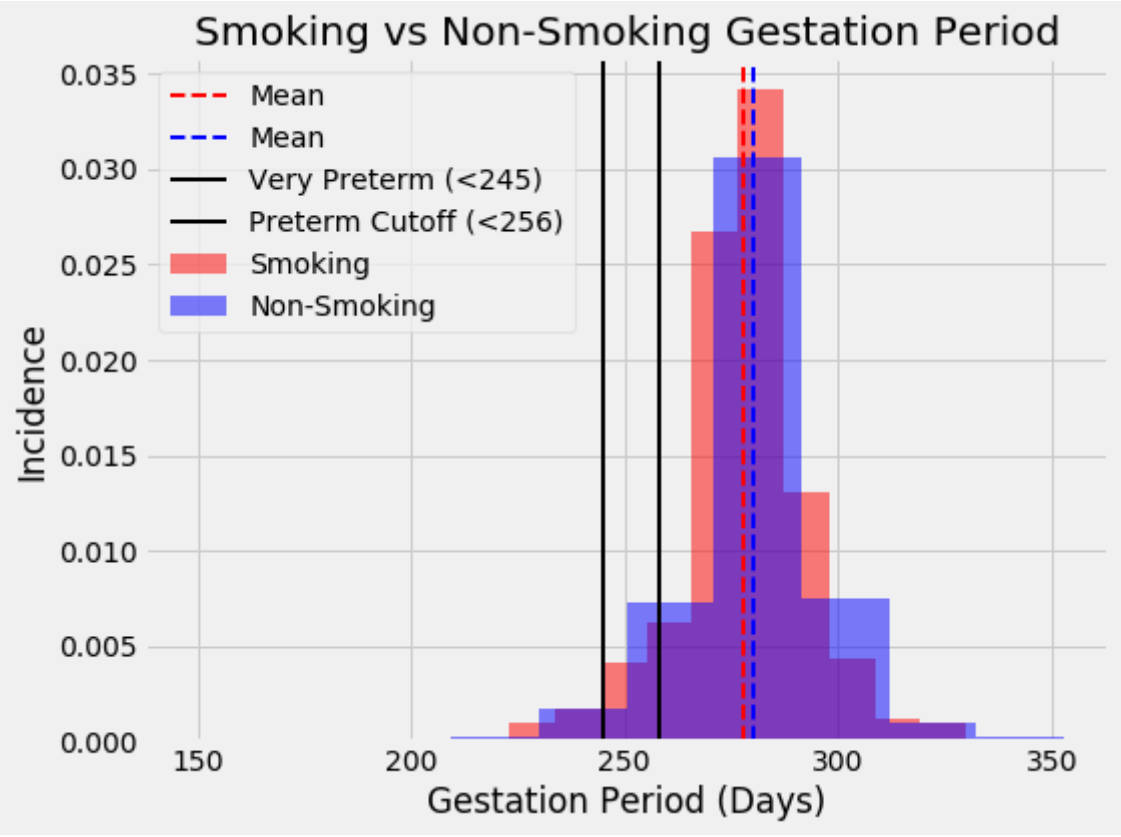
In [17]:
```
plt.figure(figsize=(8,6))

# Histograms of smoking/nonsmoking birth weights
plt.hist(smoke_df['bwt'], label='Smoking', color='red', alpha=0.5, normed=Tr
plt.hist(no_smoke_df['bwt'], label='Non-Smoking', color='blue', alpha=0.5, n

# Lines for mean of birth weights
line_kwargs = {'linewidth' : 2, 'linestyle' :'dashed'}
plt.axvline(**line_kwargs, x=np.mean(smoke_df['bwt']), color='red', label='M
plt.axvline(**line_kwargs, x=np.mean(no_smoke_df['bwt']), color='blue', labe

# Plot text
plt.legend()
plt.title('Smoking vs Non-Smoking Birth Weights')
plt.xlabel('Birth Weight (Ounces)')
plt.ylabel('Incidence')

# display plot
plt.tight_layout()
plt.savefig('bwt-histogram.png', dpi=420)
plt.show()
```



In [18]:
```
print("Count smoking %d" % (len(smoke_df['bwt'])))
print("Smoking birthweight std: %.3f" % (smoke_df['bwt'].std()))
print("Count non smoking %d" % (len(no_smoke_df['bwt'])))
print("Non Smoking birthweight std: %.3f" % (no_smoke_df['bwt'].std()))
```

```
Count smoking 479
Smoking birthweight std: 18.180
Count non smoking 732
Non Smoking birthweight std: 17.353
```

In [19]:
```
#
#
#Any value 246-258 we will consider and preterm
#Any Value <246 we will consider Very Preterm
#Any Value >258 is “normal”

plt.figure(figsize=(8,6))

# Histograms of smoking/nonsmoking birth weights
plt.hist(smoke_df['gestation'], label='Smoking', color='red', alpha=0.5, nor
plt.hist(no_smoke_df['gestation'], label='Non-Smoking', color='blue', alpha=

# Lines for mean of birth weights
line_kwargs = {'linewidth' : 2, 'linestyle' :'dashed'}
plt.axvline(**line_kwargs, x=np.mean(smoke_df['gestation']), color='red', la
plt.axvline(**line_kwargs, x=np.mean(no_smoke_df['gestation']), color='blue'

plt.axvline(linewidth=2, x=245, label='Very Preterm (<245)', color='black')
plt.axvline(linewidth= 2, x=258, label='Preterm Cutoff (<256)', color='black

# Plot text
plt.legend()
plt.title('Smoking vs Non-Smoking Gestation Period')
plt.xlabel('Gestation Period (Days)')
plt.ylabel('Incidence')

# display plot
plt.tight_layout()
plt.savefig('gestation-histogram.png', dpi=420)
plt.show()
```



In [20]:
```
plt.boxplot([smoke_df['bwt'], no_smoke_df['bwt']], labels=['Smoke','Non-Smok
plt.ylabel('Weight (Ounces)')
plt.title('Smoking vs Non Smoking Birth Weights')

# display plot
plt.tight_layout()
plt.savefig('bwt-boxplot.png', dpi=420)
plt.show()
```

Question 3

Compare the frequency, or incidence, of low-birth-weight babies for the two groups. How reliable do you think your estimates are? That is, how would the incidence of low birth weight change if a few more or fewer babies were classified as low birth weight?

In [ ]:
```
```

OMEGALUL

In [21]:
```
text = '../test.txt'
a = pd.read_csv(text, sep=r"\s*", header=None)
```

```
/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/ipykernel/__main__.py:2: ParserWarning: Falling back to the 'python'
engine because the 'c' engine does not support regex separators (separato
rs > 1 char and different from '\s+' are interpreted as regex); you can a
void this warning by specifying engine='python'.
  from ipykernel import kernelapp as app

---------------------------------------------------------------------------
--
FileNotFoundError                         Traceback (most recent call las
t)
<ipython-input-21-5acd2a4dccff> in <module>()
      1 text = '../test.txt'
----> 2 a = pd.read_csv(text, sep=r"\s*", header=None)

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/pandas/io/parsers.py in parser_f(filepath_or_buffer, sep, delimiter,
 header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dt
ype, engine, converters, true_values, false_values, skipinitialspace, ski
prows, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_
lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, da
yfirst, iterator, chunksize, compression, thousands, decimal, linetermina
tor, quotechar, quoting, escapechar, comment, encoding, dialect, tupleize
_cols, error_bad_lines, warn_bad_lines, skipfooter, skip_footer, doublequ
ote, delim_whitespace, as_recarray, compact_ints, use_unsigned, low_memor
y, buffer_lines, memory_map, float_precision)
    653                     skip_blank_lines=skip_blank_lines)
    654
--> 655         return _read(filepath_or_buffer, kwds)
    656
    657     parser_f.__name__ = name

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/pandas/io/parsers.py in _read(filepath_or_buffer, kwds)
    403
    404     # Create the parser.
--> 405     parser = TextFileReader(filepath_or_buffer, **kwds)
    406
    407     if chunksize or iterator:

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/pandas/io/parsers.py in __init__(self, f, engine, **kwds)
    762             self.options['has_index_names'] = kwds['has_index_nam
es']
    763
--> 764         self._make_engine(self.engine)
    765
    766     def close(self):

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/pandas/io/parsers.py in _make_engine(self, engine)
    993                                     ' "c", "python", or' ' "python-f
wf")'.format(
    994                                         engine=engine))
```

```
--> 995             self._engine = klass(self.f, **self.options)
    996
    997     def _failover_to_python(self):

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/pandas/io/parsers.py in __init__(self, f, **kwds)
   1982         f, handles = _get_handle(f, 'r', encoding=self.encoding,
   1983                                  compression=self.compression,
-> 1984                                  memory_map=self.memory_map)
   1985         self.handles.extend(handles)
   1986

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pack
ages/pandas/io/common.py in _get_handle(path_or_buf, mode, encoding, comp
ression, memory_map, is_text)
    383         elif is_text:
    384             # Python 3 and no explicit encoding
--> 385             f = open(path_or_buf, mode, errors='replace')
    386         else:
    387             # Python 3 and binary mode

FileNotFoundError: [Errno 2] No such file or directory: '../test.txt'
```

In [ ]:
```
a.head()
```

In [ ]:
```
g = df1[df1.weight != 999]
g = g[g.gestation != 999]
g = g[g.age != 99]
g = g[g.smoke != 9]
g = g[g.height != 99]
g = g[g.bwt != 999]
```

In [ ]:
```
for c in g.columns:
    print(max(g[c]))
```

In [ ]:
```
g.columns
```

In [ ]:
```
#Any value 246-258 we will consider and preterm
#Any Value <246 we will consider Very Preterm
#Any Value >258 is “normal”
v_pre_g = g[g.gestation < 246]
norm_g = g[g.gestation > 258]
pre_g = g[g.gestation >= 246]
pre_g = pre_g[pre_g.gestation <= 258]
```

In [ ]:
```
vp_mean = np.mean(v_pre_g['bwt'])
p_mean = np.mean(pre_g['bwt'])
n_mean = np.mean(norm_g['bwt'])
```

In [ ]:
```
plt.bar(x=[vp_mean, p_mean, n_mean])
```

In [ ]:
```
N = 3
menMeans = [vp_mean, p_mean, n_mean]

fig, ax = plt.subplots()


ind = np.arange(N)    # the x locations for the groups
width = 0.35         # the width of the bars
p1 = ax.bar(ind, menMeans, width)

ax.set_title('Birth Weight by Gestational Periods')
ax.set_xticks(ind + width / 2)
ax.set_xticklabels(('Very Preterm', 'Preterm', 'Normal'))

ax.autoscale_view()

plt.ylabel('Birth Weight (oz)')
plt.tight_layout()
plt.savefig('bwt_gestation.png', dpi=420, )

plt.show()
```

In [ ]:
```
plt.title('Distribution of Birth Weights')
plt.hist(g['bwt'], label='Smoking', color='red', alpha=.8)
plt.ylabel('Freq')
plt.xlabel('Birth Weight (oz)')
plt.tight_layout()
plt.savefig('birth weight', dpi=420)
plt.show()
```

In [ ]:
```
kurtosis(g['bwt'])
```

In [ ]:
```
levene(pre_g['bwt'], v_pre_g['bwt'], norm_g['bwt'])
```

In [ ]:
```
from scipy.stats import f_oneway
f_oneway(pre_g['bwt'], v_pre_g['bwt'], norm_g['bwt'])
```

In [ ]:
```
from scipy.stats import kruskal
kruskal(pre_g['bwt'], v_pre_g['bwt'], norm_g['bwt'])
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [71]:
```
low_bwt = 88.18
low_bwts = [88.18 - 10,88.18 -5,88.18,88.18 +5,88.18 +10]
diffs = []
```

In [72]:
```
for low_bwt in low_bwts:
    no_smoke_low_bwt = no_smoke_df[no_smoke_df.bwt < low_bwt]
    no_smoke_hi_bwt = no_smoke_df[no_smoke_df.bwt >= low_bwt]

    smoke_low_bwt = smoke_df[smoke_df.bwt < low_bwt]
    smoke_hi_bwt = smoke_df[smoke_df.bwt >= low_bwt]

    smoke_rate = (len(smoke_low_bwt) / (len(smoke_hi_bwt) + len(smoke_low_bw
    no_smoke_rate=(len(no_smoke_low_bwt) / (len(no_smoke_hi_bwt) + len(no_sm
    diffs.append(smoke_rate- no_smoke_rate)
```

In [73]:
```
No Smoking low birth rate: %.2f percent" % (len(no_smoke_low_bwt) / (len(no_
```

```
No Smoking low birth rate: 6.42 percent
```

In [74]:
```
print("Smoking low birth rate: %.2f" % (len(smoke_low_bwt) / (len(smoke_hi_b
```

```
Smoking low birth rate: 19.00
```

In [76]:
```
low_bwts
```

Out[76]:
```
[78.18, 83.18, 88.18, 93.18, 98.18]
```

In [75]:
```
diffs
```

Out[75]:
```
[1.7654037897714956,
 2.608177327538018,
 5.345266208060966,
 7.600362777644684,
 12.577147290005364]
```

In [ ]:

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from scipy.stats import ttest_ind
from scipy.stats.mstats import mquantiles
from scipy.stats import kurtosis
from scipy.stats import probplot
```

In [3]:
```
# Set up plot styles
plt.style.use(['fivethirtyeight'])
```

In [4]:
```
df1 = pd.read_csv('babies.txt', delim_whitespace=True)
df2 = pd.read_csv('babies23.txt', delim_whitespace=True)
```

In [5]:
```
df1 = df1[df1.weight != 999]
min(df1['weight'])
```

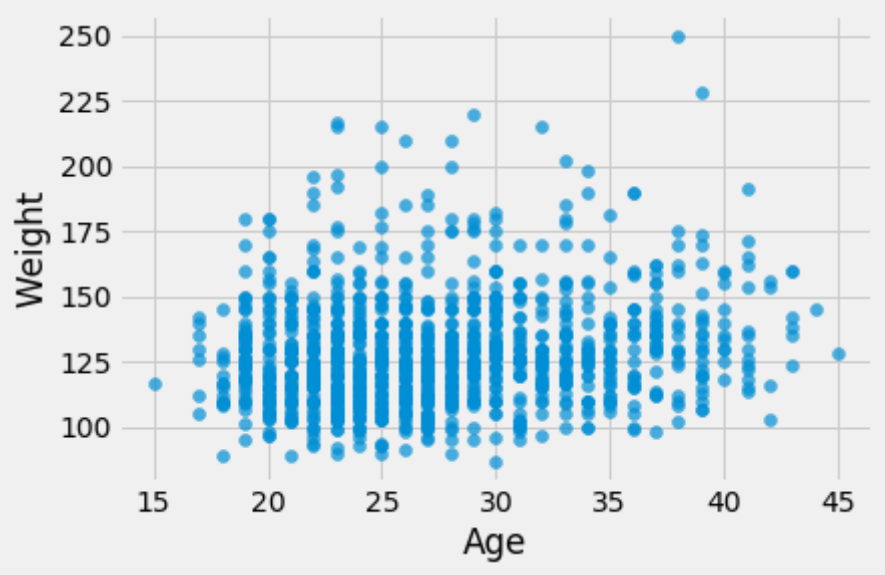
Out[5]: 87

In [6]:
```
df1.head()
```

Out[6]:

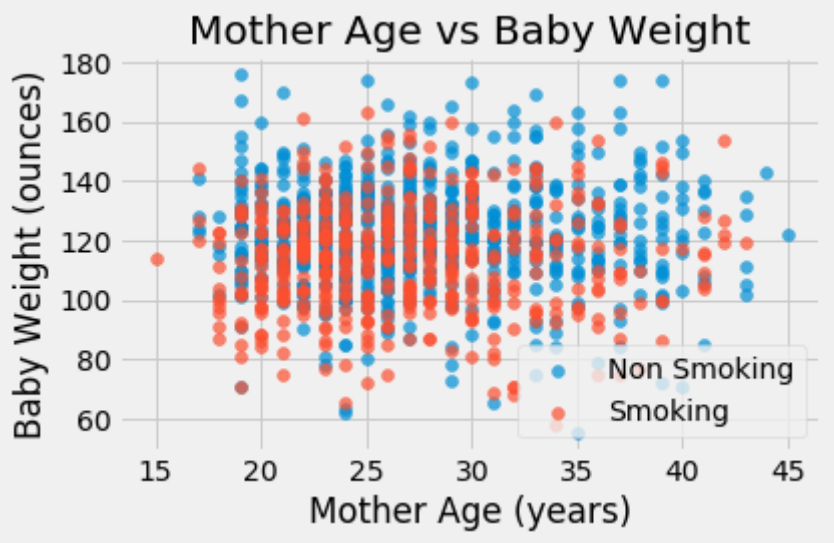
	bwt	gestation	parity	age	height	weight	smoke
0	120	284	0	27	62	100	0
1	113	282	0	33	64	135	0
2	128	279	0	28	64	115	1
3	123	999	0	36	69	190	0
4	108	282	0	23	67	125	1

In [7]:
```
df1 = df1[(df1['weight'] < 999) & (df1['age'] < 99)]

age = np.array(df1['age'])
weight = np.array(df1['weight'])

plt.scatter(age, weight, alpha=0.7)
plt.xlabel('Age')
plt.ylabel('Weight')
plt.show()
```



In [8]:
```
bwt_ns_1= list(df1[df1['smoke'] == 0]['bwt'])
bwt_s_1 = list(df1[df1['smoke'] == 1]['bwt'])
```

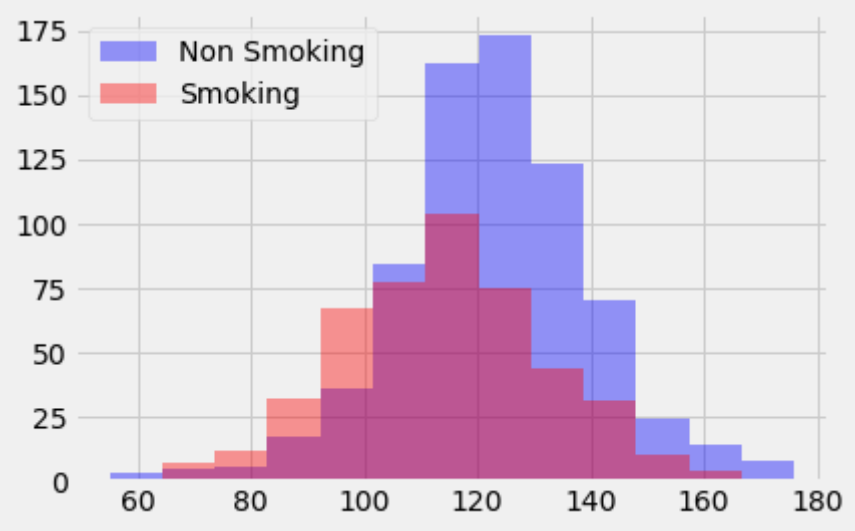
In [13]:
```
age = np.array(df1[df1['smoke'] == 0]['age'])
weight = np.array(df1[df1['smoke'] == 0]['bwt'])
plt.scatter(age, weight, alpha=0.7, label='Non Smoking')

age = np.array(df1[df1['smoke'] == 1]['age'])
weight = np.array(df1[df1['smoke'] == 1]['bwt'])
plt.scatter(age, weight, alpha=0.7, label='Smoking')

plt.xlabel('Mother Age (years)')
plt.ylabel('Baby Weight (ounces)')
plt.title('Mother Age vs Baby Weight')
plt.legend()
plt.tight_layout()
plt.savefig('scatter.png', dpi=420, )
plt.show()
```



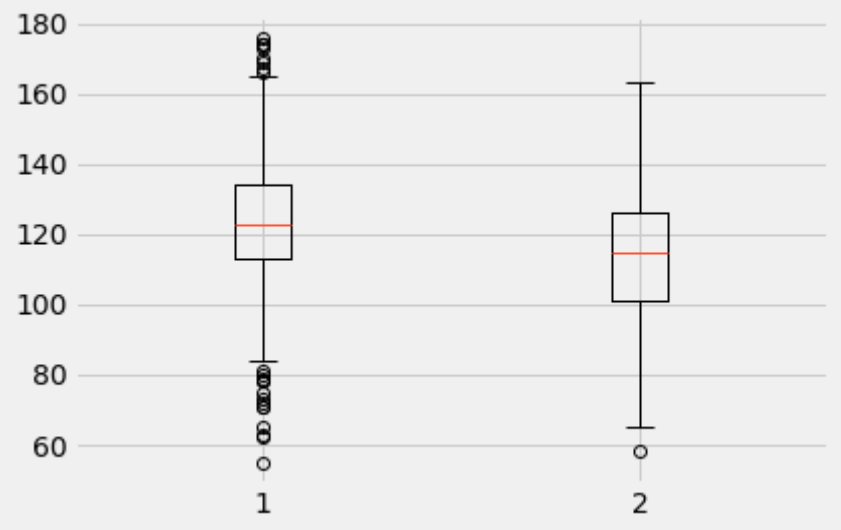
In [14]:
```
bwt_ns = bwt_ns_1
bwt_s = bwt_s_1
```

Get hist arrays

In [15]:
```
_, bins = np.histogram(np.stack([bwt_ns, bwt_ns]), bins=13)

plt.hist(bwt_ns, label='Non Smoking', bins=bins, alpha=0.4, color='blue')
plt.hist(bwt_s, label='Smoking', bins=bins, alpha=0.4, color='red')
plt.legend()
plt.show()
```



In [16]:
```
plt.boxplot([bwt_ns, bwt_s], )
plt.show()
```



In [17]:
```
print(np.mean(bwt_ns))
print(np.var(bwt_ns))
```

```
122.95724137931035
305.0698958382878
```

In [18]:
```
print(np.mean(bwt_s))
print(np.var(bwt_s))
```

```
113.9396551724138
334.4015309155767
```

In [19]:
```
bwt_ns_sample = np.random.choice(bwt_ns, 2000, replace=True)
bwt_s_sample = np.random.choice(bwt_ns, 2000, replace=True)

ttest_ind(bwt_ns_sample, bwt_s_sample, equal_var=True)
```

Out[19]:
```
Ttest_indResult(statistic=0.3309014611189214, pvalue=0.7407362591131517)
```

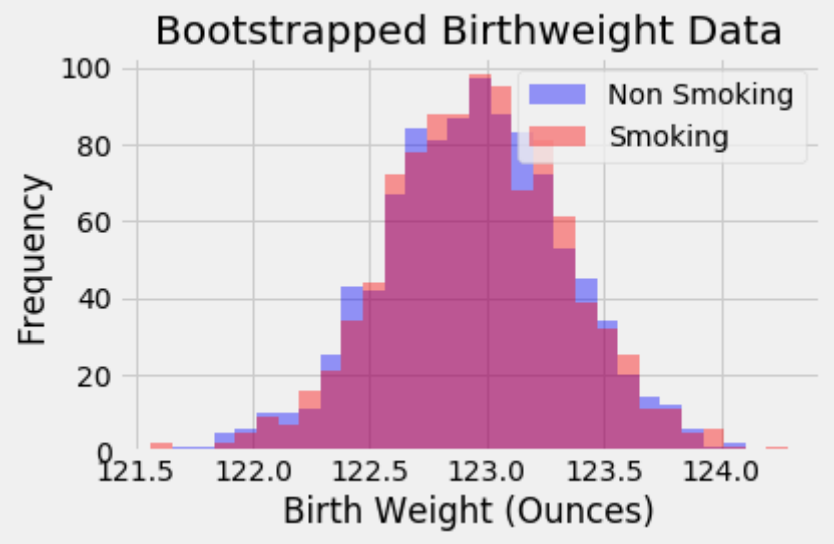
In [20]:
```
len(bwt_ns)
```

Out[20]:
```
725
```

In [21]:
```
len(bwt_s)
```

Out[21]:
```
464
```

In [31]:
```
ns_means = []
s_means = []

for i in range(1000):
    ns_means.append(np.mean(np.random.choice(bwt_ns, 2000, replace=True)))
    s_means.append(np.mean(np.random.choice(bwt_ns, 2000, replace=True)))

_, bins = np.histogram(np.stack([ns_means, s_means]), bins=30)
plt.hist(ns_means, label='Non Smoking', bins=bins, alpha=0.4, color='blue')
plt.hist(s_means, label='Smoking', bins=bins, alpha=0.4, color='red')
plt.legend()


plt.title('Bootstrapped Birthweight Data')
plt.xlabel('Birth Weight (Ounces)')
plt.ylabel('Frequency')

plt.tight_layout()
# display plot
plt.savefig('bootstrap.png', dpi=420)
plt.show()

print(kurtosis(ns_means, fisher=False))
print(kurtosis(s_means, fisher=False))
```




```
3.014079569854474
3.171704309472756
```

In [23]:
```
mquantiles(bwt_ns)
```

Out[23]:
```
array([113., 123., 134.])
```

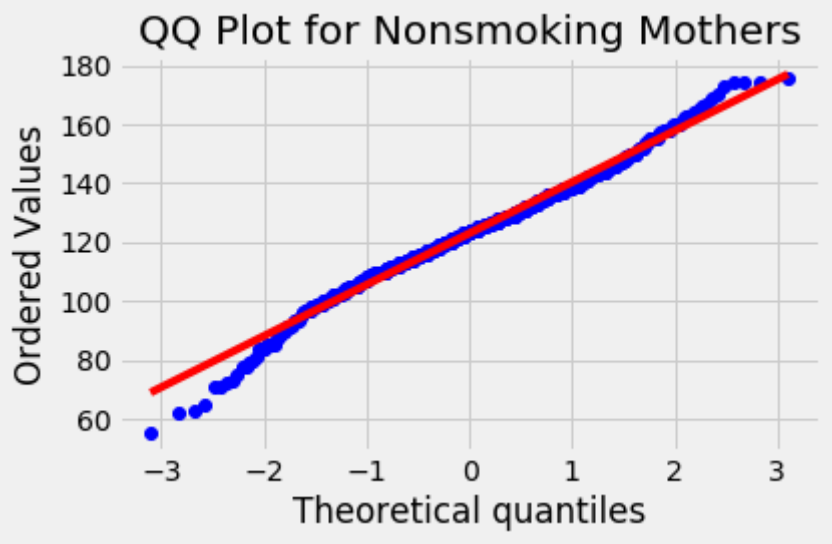
In [14]:
```
mquantiles(bwt_s)
```

Out[14]:
```
array([102., 115., 126.])
```

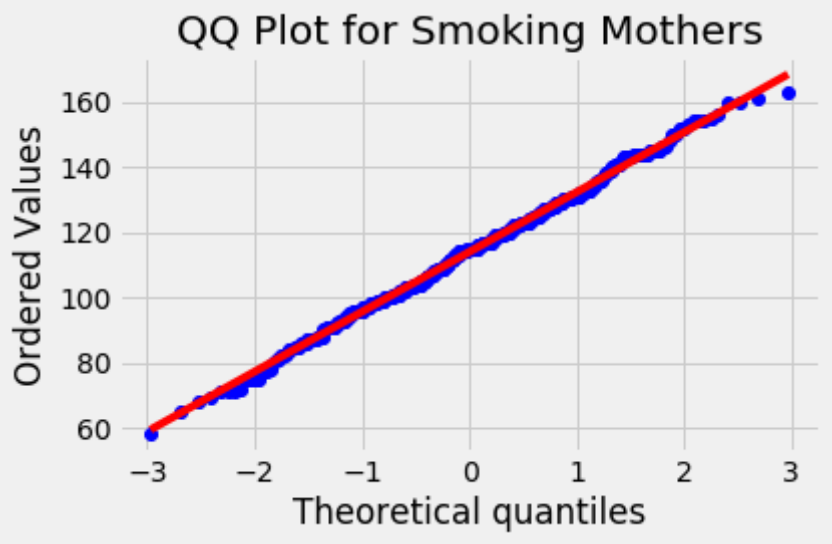
In [15]:
```
print(kurtosis(bwt_ns, fisher=False))
print(kurtosis(bwt_s, fisher=False))
```

```
4.037060312433822
2.988032478793404
```

In [34]:
```
probplot(bwt_ns, plot=plt)

plt.title('QQ Plot for Nonsmoking Mothers')

plt.tight_layout()
# display plot
plt.savefig('qq_no_smoking.png', dpi=420)

plt.show()
```



In [38]:
```
probplot(bwt_s, plot=plt)

plt.title('QQ Plot for Smoking Mothers')

plt.tight_layout()
# display plot
plt.savefig('qq_smoking.png', dpi=420)

plt.show()
```



In [ ]:
```

```