In [79]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
import statsmodels.api as sm
from statsmodels.stats.proportion import proportions_ztest
import pylab
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Math 189 HW1 smoking vs. pregancy

Yu-Chun Chen, A13356506

Yanyu Tao, A13961185

Bolin Yang, A92111272

Shuibenyang Yuan, A14031016

Haoming Zhang, A14012520

Mingxuan Zhang, A13796895

This project researches the difference between the babie' weights of smoker mother and those of non-smoker mother

Reading Data & Cleaning Data

In [80]:
```
temp = open('babies23-17em2pa-1r6aufk.txt').read().splitlines()
templist = []
for i in temp:
    templist += [i.split(' ')]
data_2_raw = []
for i in templist:
    lst = []
    for j in i:
        if j != '':
            lst += [j]
    data_2_raw += [lst]
    lst = []
```

In [81]:
```
data_2 = pd.DataFrame(data_2_raw[1:], columns = data_2_raw[0])
```

In [82]:
```
data_2 = data_2.applymap(lambda x: pd.to_numeric(x))
data_2.columns = ['id', 'pluralty', 'outcome', 'date', 'gestation', 'sex',
       'race', 'age', 'ed', 'ht', 'wt2', 'drace', 'dage', 'ded', 'dht', 'dw
       'marital', 'inc', 'smoke', 'time', 'number']
```

Comparison Set 1

Non-smoker Mothers vs. Mothers who smoke at pregancy

(below will be denoted as Non-Smokers & Smokers for simplicity)

In [51]:
```
non_smoking = data_2.loc[(data_2['smoke'] == 0) |(data_2['smoke'] == 3) | (
smoking = data_2.loc[(data_2['smoke'] > 0) & (data_2['smoke'] < 2)]
baby_weights = pd.DataFrame(
        {
            "Babies' Birth Weights born to Non-smoking in pregancy (in ounc
            "Babies' Birth Weights born to pregant smoking mothers (in ounc
        }
)
baby_weights
```

Out[51]:

	Babies' Birth Weights born to Non-smoking in pregancy (in ounce)	Babies' Birth Weights born to pregant smoking mothers (in ounce)
count	742.000000	484.000000
mean	123.047170	114.109504
std	17.398689	18.098946
min	55.000000	58.000000
25%	113.000000	102.000000
50%	123.000000	115.000000
75%	134.000000	126.000000
max	176.000000	163.000000

In [52]:
```
smoking['wt1'].skew()
```

Out[52]: -0.033699506713282625

In [53]:
```
non_smoking['wt1'].skew()
```

Out[53]: -0.18736306526595664

In [54]:
```
stats.kurtosis(smoking['wt1'], fisher = False)
```

Out[54]: 2.988032478793404

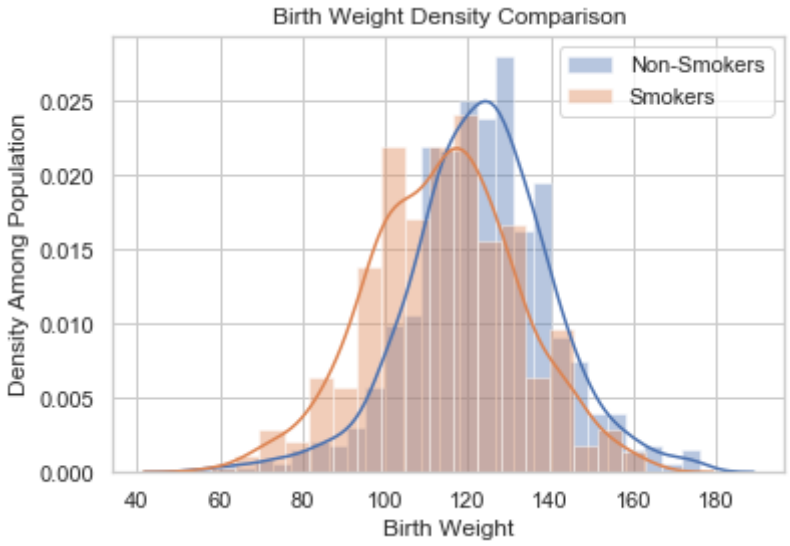
In [55]:
```
stats.kurtosis(non_smoking['wt1'], fisher = False)
```

Out[55]: 4.037060312433822

Graphical Comparison

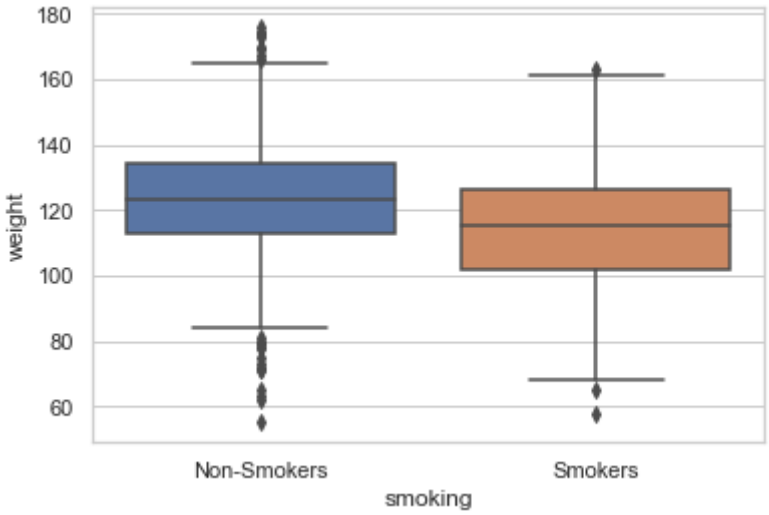
In [56]:
```
sns.distplot(non_smoking['wt1'], label="Non-Smokers")
sns.distplot(smoking['wt1'], label="Smokers")
plt.legend()
plt.title('Birth Weight Density Comparison')
plt.xlabel('Birth Weight')
plt.ylabel('Density Among Population')
```

Out[56]:
```
Text(0, 0.5, 'Density Among Population')
```

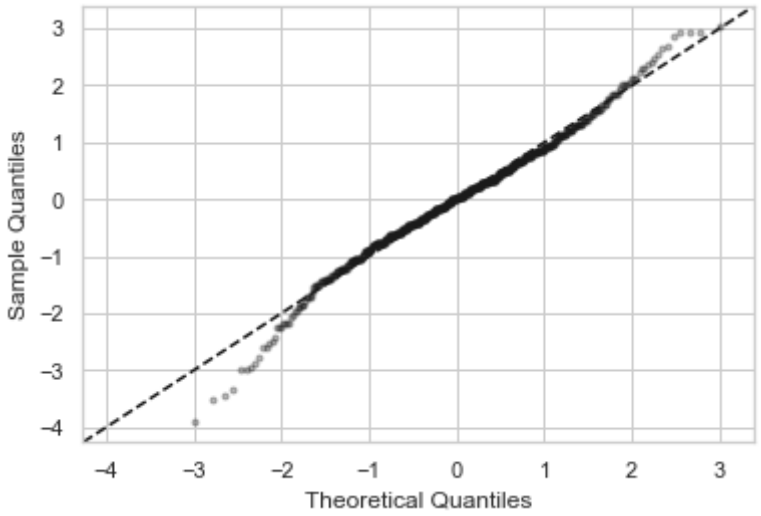


In [57]:
```
grouped_df = pd.DataFrame(
             {
                 'weight' : non_smoking['wt1'],
                 'smoking': ['Non-Smokers' for i in non_smoking['wt1']]
             }
)
grouped_df1 = pd.DataFrame(
             {
                 'weight' : smoking['wt1'],
                 'smoking': ['Smokers' for i in smoking['wt1']]
             }
)
cdf = pd.concat([grouped_df, grouped_df1])
sns.boxplot(x ='smoking', y ='weight', data = cdf)
```

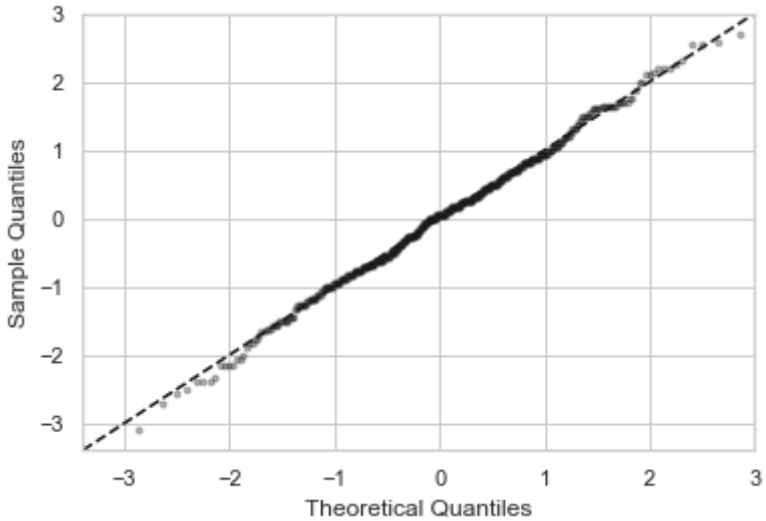
Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x116596710>



In [58]:
```
#QQ plot for non smoking
test = non_smoking['wt1']
pp = sm.ProbPlot(test, fit=True)
qq = pp.qqplot(marker='.', markerfacecolor='k', markeredgecolor='k', alpha=
sm.qqline(qq.axes[0], line='45', fmt='k--')
plt.show()
```



In [59]:
```
#QQ plot for smoking
test = smoking['wt1']
pp = sm.ProbPlot(test, fit=True)
qq = pp.qqplot(marker='.', markerfacecolor='k', markeredgecolor='k', alpha=
sm.qqline(qq.axes[0], line='45', fmt='k--')
plt.show()
```



Data Validation

We found non-smoking group's data is not normally distributed. We will process a normality check for our smoking at pregnant birth weight data set and non-smoker babies' birth weight group via Chi-Square GOF test

In [60]:
```
k2, p = stats.normaltest(non_smoking['wt1'])

alpha = 1e-3
print("p = {:g}".format(p))
if p < alpha:
    print("The null hypothesis can be rejected,")
else:
    print("The null hypothesis cannot be rejected,  the dataset is normally
```

```
p = 2.50963e-05
The null hypothesis can be rejected,
```

In [62]:
```
k2, p = stats.normaltest(smoking['wt1'])

alpha = 1e-3
print("p = {:g}".format(p))
if p < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected,")
else:
    print("The null hypothesis cannot be rejected,  the dataset is normally
```

```
p = 0.948874
The null hypothesis cannot be rejected,  the dataset is normally distribu
ted
```

T test

test the mean difference of two data sets

In [63]:
```
print('p-val is', stats.ttest_ind(non_smoking['wt1'], smoking['wt1'], equal
```

```
p-val is 3.5241301163155265e-17
```

Low Weight cut-off point w/ Chi-square independence Test

https://www.chop.edu/conditions-diseases/low-birthweight (https://www.chop.edu/conditions-diseases/low-birthweight), The weight is compared with the baby's gestational age and recorded in the medical record. A birthweight less than 2,500 grams (88.1849 ounces) is diagnosed as low birthweight. Babies weighing less than 1,500 grams (52.91094 ounces) at birth are considered very low birthweight.

cut-off point validation by chi-square test

In [66]:
```
def catogorize(non_smoking, smoking, low_weight):
    non_smoking_nlow = len([i for i in non_smoking if i <= low_weight])
    smoking_nlow = len([i for i in smoking if i <= low_weight])
    non_smoking_nn = len([i for i in non_smoking if i > low_weight])
    smoking_nn = len([i for i in smoking if i > low_weight])
    output = pd.DataFrame({
            'low weight': [non_smoking_nlow, smoking_nlow],
            'normal weight:': [non_smoking_nn, smoking_nn]}, index = ['non
    )
    return output
```

In [84]:
```
lst = []
pvals = []
for i in range(75,116, 5):
    contingency_table = catogorize(non_smoking['wt1'], smoking['wt1'], i)
    f_obs = np.array([contingency_table.iloc[0][0:].values,
                      contingency_table.iloc[1][0:].values])
    lst += [i]
    pvals += [stats.chi2_contingency(f_obs)[0:3][1]]
output = pd.DataFrame(
        {
            'p-value': pvals
        }, index = lst
)
output.index.name = 'Cut-off point'
output.to_csv('chi-square_set1.csv')
output
```

Out[84]:

Cut-off point	p-value
75	1.483697e-01
80	1.193173e-01
85	3.455917e-02
90	2.999904e-04
95	2.950961e-07
100	1.410511e-12
105	6.779451e-16
110	4.794657e-15
115	1.442709e-13

In [72]:
```
non_smoking_n = len(non_smoking['wt1'])
non_smoking_ln = len([i for i in non_smoking['wt1'] if i <= 88.1849])
smoking_n = len(smoking['wt1'])
smoking_ln = len([i for i in smoking['wt1'] if i <= 88.1849])

non_smoking_lprop = non_smoking_ln / non_smoking_n
smoking_lprop = smoking_ln / smoking_n
```

In [73]:
```
smoking_lprop
```

Out[73]:
```
0.08264462809917356
```

In [74]:
```
non_smoking_lprop
```

Out[74]:
```
0.03099730458221024
```

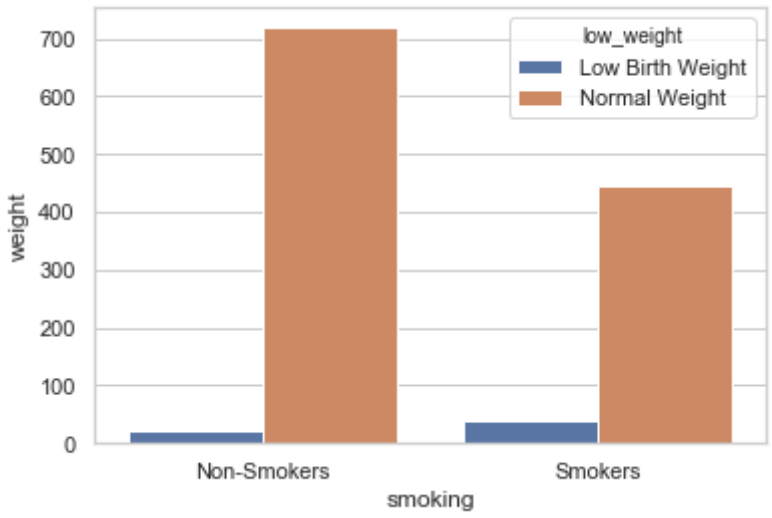
use cut-off point of 88.2 as low weight. To see the distribution plot.

low weight graph for original data set

In [75]:
```
proportion_low_weight_table = pd.DataFrame(
                            {
                                'Non-Smokers': non_smoking_lprop,
                                'Smokers': smoking_lprop
                            }
,index = ['Proportion'])
```

In [76]:
```
cdf['low_weight'] = cdf['weight'].apply(lambda x: 'Low Birth Weight' if x <
cdf1 = cdf.groupby(['low_weight','smoking']).count().reset_index()
```

In [77]:
```
sns.set(style="whitegrid")
tips = sns.load_dataset("tips")
ax = sns.barplot(x="smoking", y="weight", hue="low_weight", data=cdf1)
```



performing proportion test for original data set with cut-off lower weight 88.1849

In [78]:
```
freq = np.array([cdf1['weight'][0], cdf1['weight'][1]])
sample = np.array([cdf1['weight'][0]+cdf1['weight'][2], cdf1['weight'][1]+c
stat, pval = proportions_ztest(freq, sample)
pval
```

Out[78]: 6.236909061746852e-05

Comparison Set 2

Non smoking vs. Smoking but not in pregnancy

denoted as Non-Smokers and Smokers

In [26]:
```
non_smoking = data_2.loc[(data_2['smoke'] == 0)]
smoking = data_2.loc[(data_2['smoke'] > 1) & (data_2['smoke'] < 9)]
non_smoking.to_csv('non_smoking_all_time.csv')
smoking.to_csv('smoking_all_time.csv')
```

Stats Overview

In [27]:
```
baby_weights = pd.DataFrame(
        {
            "Babies' Birth Weights born to Non-smoking mothers (in ounce)"
            "Babies' Birth Weights born to smoking mothers (in ounce)": smo
        }
)
baby_weights
```

Out[27]:

	Babies' Birth Weights born to Non-smoking mothers (in ounce)	Babies' Birth Weights born to smoking mothers (in ounce)
count	544.000000	198.000000
mean	122.777574	123.787879
std	17.109661	18.193139
min	55.000000	62.000000
25%	113.000000	112.000000
50%	124.000000	123.000000
75%	132.250000	137.000000
max	176.000000	170.000000

In [28]:
```
smoking['wt1'].skew()
```

Out[28]: -0.3539445104412361

In [29]:
```
non_smoking['wt1'].skew()
```

Out[29]:
```
-0.12179028083493815
```

In [30]:
```
stats.kurtosis(smoking['wt1'], fisher = False)
```

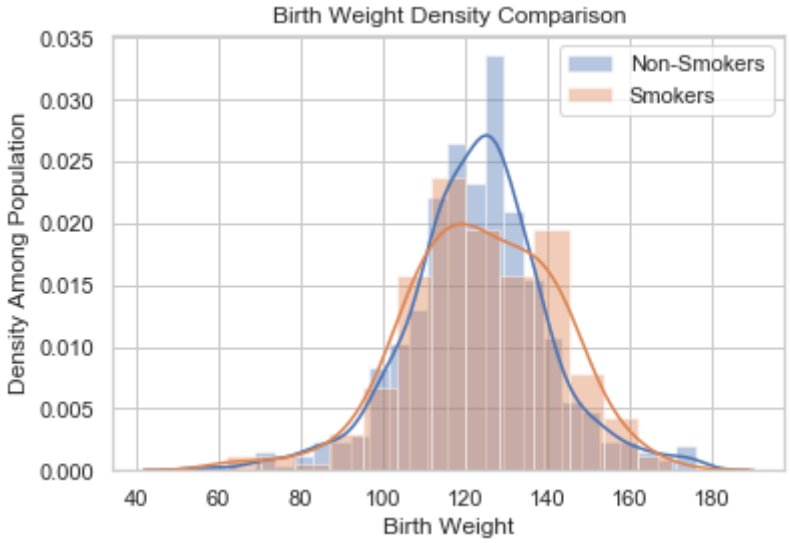
Out[30]:
```
3.4517687297202033
```

In [31]:
```
stats.kurtosis(non_smoking['wt1'], fisher = False)
```

Out[31]:
```
4.310783478561745
```

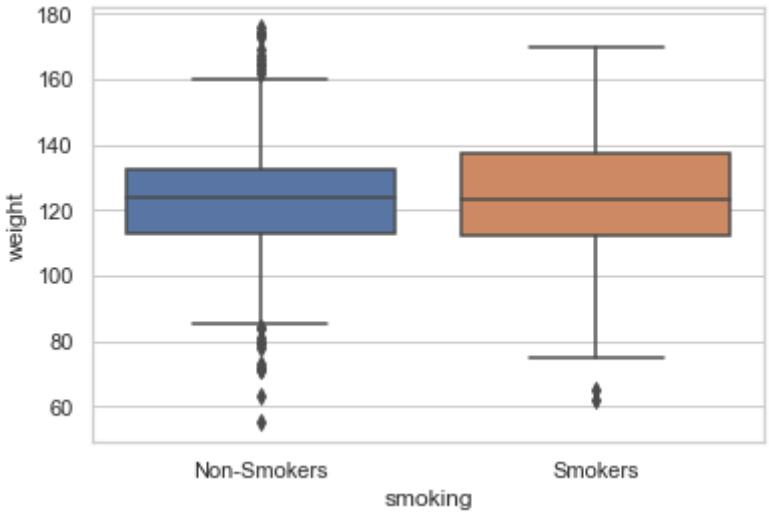
In [32]:
```
sns.distplot(non_smoking['wt1'], label="Non-Smokers")
sns.distplot(smoking['wt1'], label="Smokers")
plt.legend()
plt.title('Birth Weight Density Comparison')
plt.xlabel('Birth Weight')
plt.ylabel('Density Among Population')
```

Out[32]:
```
Text(0, 0.5, 'Density Among Population')
```



In [33]:
```
grouped_df = pd.DataFrame(
             {
                 'weight' : non_smoking['wt1'],
                 'smoking': ['Non-Smokers' for i in non_smoking['wt1']]
             }
)
grouped_df1 = pd.DataFrame(
             {
                 'weight' : smoking['wt1'],
                 'smoking': ['Smokers' for i in smoking['wt1']]
             }
)
cdf = pd.concat([grouped_df, grouped_df1])
sns.boxplot(x ='smoking', y ='weight', data = cdf)
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x116c276a0>



In [34]:
```
k2, p = stats.normaltest(non_smoking['wt1'])

alpha = 1e-3
print("p = {:g}".format(p))
if p < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected,")
else:
    print("The null hypothesis cannot be rejected, the dataset is normally
```

```
p = 8.48975e-05
The null hypothesis can be rejected,
```

In [35]:
```
k2, p = stats.normaltest(smoking['wt1'])

alpha = 1e-3
print("p = {:g}".format(p))
if p < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected,")
else:
    print("The null hypothesis cannot be rejected, the dataset is normally
```

```
p = 0.0490721
The null hypothesis cannot be rejected, the dataset is normally distribut
ed
```

T test for mean

In [36]:
```
stats.ttest_ind(non_smoking['wt1'], smoking['wt1'], equal_var = False)[1]
```

Out[36]:
```
0.49720854479542087
```

In [37]:
```
# fail to reject
```

Chi-square test to validate low weight cut-off

In [38]:
```
lst = []
pvals = []
for i in range(75,100):
    contingency_table = catogorize(non_smoking['wt1'], smoking['wt1'], i)
    f_obs = np.array([contingency_table.iloc[0][0:].values,
                      contingency_table.iloc[1][0:].values])
    lst += [i]
    pvals += [stats.chi2_contingency(f_obs)[0:3][1]]
output = pd.DataFrame(
        {
            'p-value': pvals
        }, index = lst
)
output.index.name = 'Cut-off point'
output.to_csv('chi-square_set1.csv')
output
```

88	0.862166
89	0.937247
90	0.895917
91	0.834903
92	0.924728
93	0.949936
94	0.967193
95	0.886804
96	0.972409

In [39]:
```
## Proportion
```

In [40]:
```
non_smoking_n = len(non_smoking['wt1'])
non_smoking_ln = len([i for i in non_smoking['wt1'] if i <= 88.1849])
smoking_n = len(smoking['wt1'])
smoking_ln = len([i for i in smoking['wt1'] if i <= 88.1849])

non_smoking_lprop = non_smoking_ln / non_smoking_n
smoking_lprop = smoking_ln / smoking_n
```

In [41]:
```
proportion_low_weight_table = pd.DataFrame( {
                                'Non-Smokers': non_smoking_lprop,
                                'Smokers': smoking_lprop
                            }
,index = ['Proportion'])
proportion_low_weight_table
```

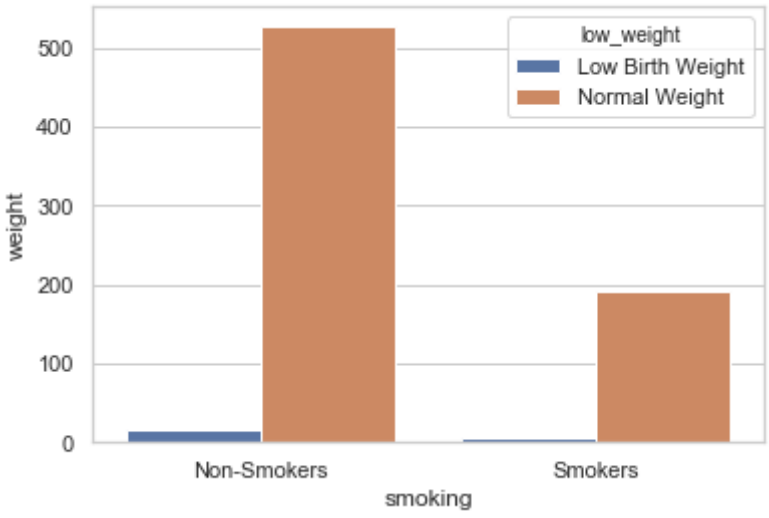
Out[41]:

	Non-Smokers	Smokers
Proportion	0.03125	0.030303

In [42]:
```
grouped_df = pd.DataFrame(
             {
                 'weight' : non_smoking['wt1'],
                 'smoking': ['Non-Smokers' for i in non_smoking['wt1']]
             }
)
grouped_df1 = pd.DataFrame(
             {
                 'weight' : smoking['wt1'],
                 'smoking': ['Smokers' for i in smoking['wt1']]
             }
)
cdf = pd.concat([grouped_df, grouped_df1])
```

In [43]:
```
cdf['low_weight'] = cdf['weight'].apply(lambda x: 'Low Birth Weight' if x <
cdf1 = cdf.groupby(['low_weight','smoking']).count().reset_index()
```

In [44]:
```
sns.set(style="whitegrid")
tips = sns.load_dataset("tips")
ax = sns.barplot(x="smoking", y="weight", hue="low_weight", data=cdf1)
```



In [45]:
```
# Proportion test for low weight
```

In [46]:
```
freq = np.array([cdf1['weight'][0], cdf1['weight'][1]])
sample = np.array([cdf1['weight'][0]+cdf1['weight'][2], cdf1['weight'][1]+c
stat, pval = proportions_ztest(freq, sample)
pval
```

Out[46]:
```
0.9475110159453997
```

In [93]:
```
# fail to reject, they are the same
```

In [ ]:
```

```