In [79]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import norm
from scipy.stats import linregress
from sklearn.utils import resample
import seaborn as sns
import statsmodels.api as sm
import pylab
```

In [80]:

```
raw=np.loadtxt('babies.txt', skiprows=1)
```

In [81]:

```
ta1=pd.DataFrame(raw)
```

In [82]:

```
ta1.columns=['bwt', 'gestation', 'parity', 'age', 'height', 'weight', 'smoke']
```

In [83]:

```
#Detect outliers and delete the data
ta1=ta1.drop(ta1[ta1['weight']==999].index)
ta1=ta1.drop(ta1[ta1['smoke']==9].index)
ta1=ta1.drop(ta1[ta1['gestation']==999].index)
ta1=ta1.drop(ta1[ta1['height']==99].index)
ta1=ta1.drop(ta1[ta1['age']==99].index)
```

In [84]:

```
#Add New column BMI
bmi=ta1['weight']*0.45/(ta1['height']*0.025)**2
```

In [85]:

```
ta1['BMI index']=bmi
```

In [86]:

```
#Drop BMI index > 18.5
ta1=ta1[ta1['BMI index']>18.5]
ta1
```

...

In [87]:

```
stat=ta1.describe()
stat[['bwt','gestation']]
```

Out[87]:

	bwt	gestation
count	1130.000000	1130.000000
mean	119.623009	279.212389
std	18.344586	16.087643
min	55.000000	148.000000
25%	109.000000	272.000000
50%	120.000000	280.000000
75%	131.000000	288.000000
max	176.000000	353.000000

In [88]:

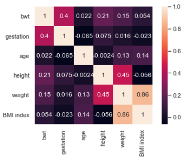
```
t=ta1.drop(['smoke','parity'],1)
```

In [89]:

```
cor=t.corr()
sns.heatmap(cor,square=True,annot=True)
```

Out[89]:

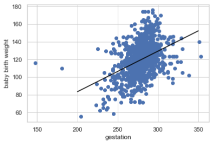
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d9d289b5f8>
```



Pearson correlation formula: $r = \frac{N\sum XY-(\sum X \sum Y)}{\sqrt{(N\sum x^2-(\sum x)^2)(N\sum y^2-(\sum y)^2\ )}}$

In [90]:

```
plt.scatter(ta1['gestation'],ta1['bwt'])
plt.xlabel('gestation')
plt.ylabel('baby birth weight')
r=linregress(ta1['gestation'],ta1['bwt'])
x=np.linspace(200,350)
a=r.slope
b=r.intercept
plt.plot(x,a*x+b,color='black')
plt.savefig('ges-bwt.jpg')
```



In [91]:

```
#Boxplot of relation between smoke and bwt
sns.set(style='whitegrid')
```

In [92]:

```
ta1_c=ta1[ta1['age']<=40]
ta1_c=ta1_c[ta1_c['age']>=20]
```

In [93]:

```
#Histogram and distribution simulation of birth weight, smoke=1
smoke=ta1_c[ta1_c['smoke']==1]
Non_smoke=ta1_c[ta1_c['smoke']==0]
smoke_bwt=smoke['bwt']
Non_smoke_bwt=Non_smoke['bwt']
```

In [94]:

```
print(smoke.count())
print(Non_smoke.count())
```

```
bwt          400
gestation    400
parity       400
age          400
height       400
weight       400
smoke        400
BMI index    400
dtype: int64
bwt          638
gestation    638
parity       638
age          638
height       638
weight       638
smoke        638
BMI index    638
dtype: int64
```

In [16]:

```
print('smoke: ',stats.mode(smoke_bwt))
print('Non-smoke: ',stats.mode(Non_smoke_bwt))
```

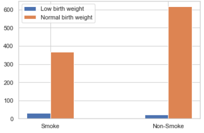
```
smoke:  ModeResult(mode=array([115.]), count=array([18]))
Non-smoke:  ModeResult(mode=array([125.]), count=array([21]))
```

In [73]:

```
lbw_s=smoke[smoke['bwt']<88.2]['bwt'].count()
bw_s=smoke[smoke['bwt']>=88.2]['bwt'].count()

lbw_ns=Non_smoke[Non_smoke['bwt']<88.2]['bwt'].count()
bw_ns=Non_smoke[Non_smoke['bwt']>=88.2]['bwt'].count()
q1=[]
q1.append(lbw_s)
q1.append(lbw_ns)

q2=[]
q2.append(bw_s)
q2.append(bw_ns)
width=0.2

index=np.arange(2)
```

In [75]:

```

fig, ax = plt.subplots()
ax.bar(index,q1,width,label="Low birth weight")
ax.bar(index+width, q2, width,label='Normal birth weight')
ax.set_xticks(index + width / 2)
ax.set_xticklabels(('Smoke','Non-Smoke'))
ax.legend()
plt.savefig('Low birth weight vs Normal birth weight')
```



In [15]:
```
#Histogram and distribution simulation of birth weight, smoke=0
fig=plt.figure(figsize=(10,10))

ax1=fig.add_subplot(221)
sns.distplot(smoke['bwt'],ax=ax1)
ax1.set_xlabel('birthweight')
ax1.set_title('smoking mother')

ax2=fig.add_subplot(222)
sns.distplot(Non_smoke['bwt'],ax=ax2,color='y')
ax2.set_xlabel('birthweight')
ax2.set_title('Non-smoking mother')

ax3=fig.add_subplot(223)
sns.boxplot(x='smoke',y='bwt',data=ta1,ax=ax3,hue='smoke')
ax3.set_ylabel('birthweight')

ax4=fig.add_subplot(224)
sns.distplot(smoke_bwt,ax=ax4,label='smoke')
sns.distplot(Non_smoke_bwt,ax=ax4,label='Non-smoker')
ax4.set_xlabel('birthweight')
ax4.legend(loc='upper right')

fig.savefig('Boxplot and histogram')
```

...

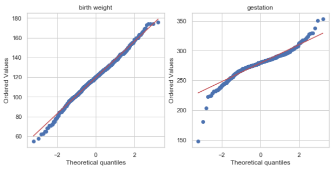
In [16]:
```
#Bootstrap calculate kurtosis and skewness
kurt0=[]
skew0=[]
kurt1=[]
skew1=[]
for i in range(1000):
    boot_smoke_bwt=resample(smoke_bwt)
    boot_non_smoke_bwt=resample(Non_smoke_bwt)
    kurt0.append(stats.kurtosis(boot_smoke_bwt,fisher=True))
    skew0.append(stats.skew(boot_smoke_bwt))
    kurt1.append(stats.kurtosis(boot_non_smoke_bwt,fisher=True))
    skew1.append(stats.skew(boot_non_smoke_bwt))
```

In [17]:
```
print('kurtosis for smoke: ', np.mean(kurt0),'\nskewness for smoke: ',np.mean(skew0))
print('kurtosis for Non-smoke: ', np.mean(kurt1),'\nskewness for Non-smoke: ',np.mean(skew1))
```

```
kurtosis for smoke:  -0.020700420828976843
skewness for smoke:  -0.026935721670656877
kurtosis for Non-smoke:  0.9895309163090449
skewness for Non-smoke:  -0.16004335769734115
```

In [19]:
```
fig=plt.figure(figsize=(10,10))
ax1=fig.add_subplot(221)
stats.probplot(ta1['bwt'],dist=norm,plot=ax1)
ax1.set_title('birth weight')
ax2=fig.add_subplot(222)
stats.probplot(ta1['gestation'],dist=norm,plot=ax2)
ax2.set_title('gestation')
```

Out[19]:
```
Text(0.5, 1.0, 'gestation')
```



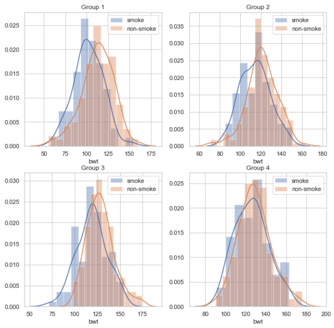
In [20]:
```
#The description of two groups
st=smoke.describe()
nst=Non_smoke.describe()
```

In [21]:
```
#Divide gestation group into 4 groups and compare the influence of smoke
#over birth weight
minimum=int(stat['gestation']['min'])
q1=int(stat['gestation']['25%'])
mean=int(stat['gestation']['50%'])
q3=int(stat['gestation']['75%'])
maximum=int(stat['gestation']['max'])

g1=ta1[ta1['gestation']<=q1]
g2=ta1[ta1['gestation']>q1]
g2=g2[g2['gestation']<=mean]
g3=ta1[ta1['gestation']>mean]
g3=g3[g3['gestation']<=q3]
g4=ta1[ta1['gestation']>q3]
```

In [22]:
```
#And we form 4 groups. For each group, we separate again the data to smoke and non-smoke
s1=g1[g1['smoke']==1]
ns1=g1[g1['smoke']==0]
s2=g2[g2['smoke']==1]
ns2=g2[g2['smoke']==0]
s3=g3[g3['smoke']==1]
ns3=g3[g3['smoke']==0]
s4=g4[g4['smoke']==1]
ns4=g4[g4['smoke']==0]
```

In [23]:
```
fig=plt.figure(figsize=(10,10))

ax1=fig.add_subplot(221)
sns.distplot(s1['bwt'],ax=ax1,label='smoke')
sns.distplot(ns1['bwt'],ax=ax1,label='non-smoke')
ax1.set_title('Group 1')
ax1.legend()

ax2=fig.add_subplot(222)
sns.distplot(s2['bwt'],ax=ax2,label='smoke')
sns.distplot(ns2['bwt'],ax=ax2,label='non-smoke')
ax2.set_title('Group 2')
ax2.legend()

ax3=fig.add_subplot(223)
sns.distplot(s3['bwt'],ax=ax3,label='smoke')
sns.distplot(ns3['bwt'],ax=ax3,label='non-smoke')
ax3.set_title('Group 3')
ax3.legend(loc='upper right')

ax4=fig.add_subplot(224)
sns.distplot(s4['bwt'],ax=ax4,label='smoke')
sns.distplot(ns4['bwt'],ax=ax4,label='non-smoke')
ax4.set_title('Group 4')
ax4.legend()

plt.savefig('Distribution over 4 groups.jpg')
```

```
C:\Users\zhijian\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an e
rror or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [24]:
```
#Define a function to bootstrap data and calculate the skewness and kurtosis
#Define normalize function
def sta(s,ns):
    kurt_s=[]
    skew_s=[]
    kurt_ns=[]
    skew_ns=[]
    group_ns=[]
    for i in range(1000):
        x_s=resample(s['bwt'])
        x_ns=resample(ns['bwt'])
        kurt_s.append(stats.kurtosis(x_s))
        kurt_ns.append(stats.kurtosis(x_ns))
        skew_s.append(stats.skew(x_s))
        skew_ns.append(stats.skew(x_ns))
    return pd.DataFrame([['kurtosis for smoke:', np.mean(kurt_s),'skewness for smoke:',np.mean(skew_s)],
           ['kurtosis for Non-smoke:', np.mean(kurt_ns),'skewness for Non-smoke:',np.mean(skew_ns)]])
def normalize(x):
    return (x-np.mean(x))/np.std(x)
```

In [25]:
```
#Group1
sta(s1,ns1)
```

Out[25]:

	0	1	2	3
0	kurtosis for smoke:	0.057575	skewness for smoke:	0.037529
1	kurtosis for Non-smoke:	0.357095	skewness for Non-smoke:	-0.424144

In [26]:
```
stats.kstest(normalize(s1['bwt']), 'norm')
```

Out[26]:
```
KstestResult(statistic=0.05195875619612467, pvalue=0.8856717569210855)
```

In [27]:
```
stats.kstest(normalize(ns1['bwt']), 'norm')
```

Out[27]:
```
KstestResult(statistic=0.06914297887844378, pvalue=0.4138208394658598)
```

```
In [30]: #perform booststrap
         bt=[]
         for i in range(1000):
             new1=resample(s1['bwt'])
             new2=resample(ns1['bwt'])
             f=stats.ttest_ind(new1,new2,equal_var=False)[1]
             bt.append(f)
```

```
In [31]: print('The mean p-value is: ', np.mean(bt))

         The mean p-value is:  0.0006985700770165726
```

```
In [67]: #Group2
         sta(s2,ns2)
```

Out[67]:

	0	1	2	3
0	kurtosis for smoke:	-0.456983	skewness for smoke:	0.151926
1	kurtosis for Non-smoke:	0.696749	skewness for Non-smoke:	-0.306987

```
In [69]: stats.kstest(normalize(s2['bwt']), 'norm')

Out[69]: KstestResult(statistic=0.07160879598278153, pvalue=0.529534536559876)
```

```
In [70]: stats.kstest(normalize(ns2['bwt']), 'norm')

Out[70]: KstestResult(statistic=0.05850244374565028, pvalue=0.5970564375272418)
```

```
In [32]: bt2=[]
         for i in range(1000):
             new1=resample(s2['bwt'])
             new2=resample(ns2['bwt'])
             f=stats.ttest_ind(new1,new2,equal_var=False)[1]
             bt2.append(f)
```

```
In [33]: print('The mean p-value is: ', np.mean(bt2))

         The mean p-value is:  0.0059093854720566955
```

```
In [128]: #Group3
          sta(s3,ns3)
```

Out[128]:

	0	1	2	3
0	kurtosis for smoke:	-0.176585	skewness for smoke:	-0.124142
1	kurtosis for Non-smoke:	0.424900	skewness for Non-smoke:	0.569729

```
In [75]: print(stats.kstest(normalize(s3['bwt']), 'norm'))
         print(stats.kstest(normalize(ns3['bwt']), 'norm'))

         KstestResult(statistic=0.060510232897182326, pvalue=0.8533512474725926)
         KstestResult(statistic=0.06625317604580283, pvalue=0.38734800406728986)
```

```
In [34]: bt3=[]
         for i in range(1000):
             new1=resample(s3['bwt'])
             new2=resample(ns3['bwt'])
             f=stats.ttest_ind(new1,new2,equal_var=False)[1]
             bt3.append(f)
```

```
In [35]: print('The mean p-value is: ', np.mean(bt3))

         The mean p-value is:  0.0006449392814525881
```

```
In [129]: #Group4
          sta(s4,ns4)
```

Out[129]:

	0	1	2	3
0	kurtosis for smoke:	-0.438254	skewness for smoke:	0.255058
1	kurtosis for Non-smoke:	0.122453	skewness for Non-smoke:	0.349934

```
In [82]: print(stats.kstest(normalize(s4['bwt']), 'norm'))
         print(stats.kstest(normalize(ns4['bwt']), 'norm'))

         KstestResult(statistic=0.05580891386345749, pvalue=0.9515974991524998)
         KstestResult(statistic=0.05198208287427053, pvalue=0.7347264326097929)
```

```
In [36]: bt4=[]
         for i in range(1000):
             new1=resample(s4['bwt'])
             new2=resample(ns4['bwt'])
             f=stats.ttest_ind(new1,new2,equal_var=False)[1]
             bt4.append(f)
```

```
In [37]: print('The mean p-value is: ', np.mean(bt4))

         The mean p-value is:  0.2604868187550974
```

```
In [27]: #Dataset 2
         rw=np.loadtxt('baby123.txt',skiprows=1)
```

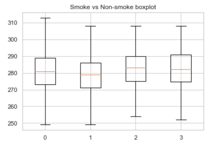
```
In [28]: ta2=pd.DataFrame(rw)
         ta2.columns=['id',  'pluralty', 'outcome', 'date', 'gestation', 'sex', 'wt', 'parity',
                      'race', 'age', 'ed', 'ht', 'wtm', 'drace', 'dage', 'ded','dht', 'dwt', 'marital',
                      'inc', 'smoke', 'time', 'number']
```

```
In [54]: ta2=ta2.drop(ta2[ta2['race']==99].index)
         ta2=ta2.drop(ta2[ta2['gestation']==999].index)
         ta2=ta2.drop(ta2[ta2['age']==99].index)
         ta2=ta2.drop(ta2[ta2['drace']==99].index)
         ta2=ta2.drop(ta2[ta2['dage']==99].index)
         ta2=ta2.drop(ta2[ta2['ded']==99].index)
         ta2=ta2.drop(ta2[ta2['dwt']==999].index)
         ta2=ta2.drop(ta2[ta2['inc']==98].index)
         ta2=ta2.drop(ta2[ta2['smoke']==9].index)
         ta2=ta2.drop(ta2[ta2['time']==9].index)
         ta2=ta2.drop(ta2[ta2['drace']==98].index)
         ta2=ta2.drop(ta2[ta2['dht']==99].index)
         ta2=ta2.drop(ta2[ta2['ed']==9].index)
```

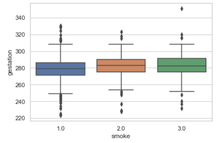
```
In [43]: ta2=ta2.drop(ta2[ta2['smoke']==9].index)
         ta2=ta2.drop(ta2[ta2['gestation']==999].index)
```

```
In [44]: ns2=ta2[ta2['smoke']!=0]
```

```
In [47]: #ta1 contains biological info of babies
         #ta2 contains mother mother info
         r0=Non_smoke['gestation']
         r1=ns2[ns2['smoke']==1]['gestation']
         r2=ns2[ns2['smoke']==2]['gestation']
         r3=ns2[ns2['smoke']==3]['gestation']

         data=[r0,r1,r2,r3]
         fig7, ax7 = plt.subplots()
         ax7.set_title('Smoke vs Non-smoke boxplot')
         ax7.boxplot(data,showfliers=False)
         plt.xticks([1, 2, 3,4], [0,1,2,3])

Out[47]: ([<matplotlib.axis.XTick at 0x24f71df3470>,
           <matplotlib.axis.XTick at 0x24f71df1cf8>,
           <matplotlib.axis.XTick at 0x24f71df1a58>,
           <matplotlib.axis.XTick at 0x24f721d34e0>],
          <a list of 4 Text xticklabel objects>)
```



```
In [46]: sns.boxplot(x='smoke',y='gestation',data=ns2)

Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x24f71cc00f0>
```



```
In [ ]:
```

```
The downloaded binary packages are in
    /var/folders/0l/v6fq50ss70l5drt02h732wrr0000gn/T//Rtmp0g4Yn0/downl
oaded_packages
> install.packages(’moments’, dependencies=TRUE)
Error in install.packages : Updating loaded packages
> library(moments)
>
> setwd(″/Users/nuochen/Desktop/Math189″)
> Data <- read.table(″babies.txt″,header = TRUE)
>
> #define BMI
> Data$BMI <- ((Data$wt*0.45)/(Data$ht*0.025)ˆ2)
>
> Data2 <- subset(Data, Data$BMI < 30)
>
> #eliminate outlier 999
> irreg.index <- which (Data2$gestation == 999)
> Data2.irregular <- Data2[irreg.index,]
> Data2.irregular
       id pluralty outcome date gestation sex  wt parity race age ed ht
wt.1 drace dage ded
4      61        5       1 1504       999   1 123      2    0  36  5
69  190     3   43   4
90   1361        5       1 1457       999   1 114      1    7  24  4
67  113     7   25   4
94   1457        5       1 1573       999   1  92      2    0  31  5
67  130     0   32   5
99   1537        5       1 1388       999   1 128      2    5  35  5
62  110     2   35   5
155  2557        5       1 1408       999   1 129      2    1  23  2
99  999     0   24   2
243  3943        5       1 1472       999   1 111      8    0  27  1
63  105     4   31   1
651  6823        5       1 1501       999   1 121      4    0  31  2
68  132     0   33   2
707  6990        5       1 1404       999   1 114      2    4  23  4
63  116     0   24   5
740  7083        5       1 1481       999   1  71      1    7  19  1
64  120     7   25   2
880  7466        5       1 1492       999   1 129      0    5  19  2
61  110     6   19   1
```

```
964  7689         5        1 1485        999   1 107       1     0  19  1
60  118     5   22   1
972  7711         5        1 1507        999   1 136      13     0  36  2
66  135     0   39   5
1193 8499         5        1 1680        999   1 124       0     7  39  2
65  228     7   38   1
     dht dwt marital inc smoke time number       BMI
4     68 197       1   8     3    5      5 18.601134
90    74 170       1   6     1    1      5 18.284696
94    99 999       1   5     3    9      1 14.756070
99    71 168       1   9     3    3      1 23.975026
155   99 999       1   4     1    1      9  9.476584
243   99 999       1   4     1    1      3 20.136054
651   70 180       1   5     0    0      0 18.840830
707   99 999       1   3     1    1      2 20.680272
740   66 160       1   1     0    0      0 12.480469
880   67 156       1   0     0    0      0 24.961032
964   99 999       1   1     0    0      0 21.400000
972   72 185       1   7     0    0      0 22.479339
1193  70 220       1   4     0    0      0 21.131361
> normal.index <- which (Data2$gestation != 999)
> Data2.normal <- Data2[normal.index,]
> boxplot(gestation~smoke,Data2.normal)
>
> #subset smoker/nonsmoker
> Data2.smoke <- subset(Data2.normal,Data2.normal$smoke == 1)
> Data2.nonsmoke <- subset(Data2.normal,
+                               (Data2.normal$smoke == 0 |
+                                Data2.normal$smoke == 2 |
+                                Data2.normal$smoke == 3 ))
> Data2.both <- subset(Data2.normal, (Data2.normal$smoke != 9))
> #total number of sample size without outliers/smoke=9
> nrow(Data2.smoke)+nrow(Data2.nonsmoke)
[1] 1205
>
> #we first want to find the mean and variance of gestation age
> summary(Data2.smoke$gestation)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  223.0   271.0   279.0   277.9   286.0   330.0
> summary(Data2.nonsmoke$gestation)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  148.0   273.0   281.0   280.1   289.0   353.0
> var(Data2.smoke$gestation)
[1] 226.7328
> var(Data2.nonsmoke$gestation)
[1] 277.5445
> sd(Data2.smoke$gestation)
[1] 15.05765
> sd(Data2.nonsmoke$gestation)
[1] 16.65967
>
> #we then find the histogram of gestation age in order to find the
distribution
> hist(Data2.smoke$gestation,
+      breaks = 100,
+      probability = TRUE,
+      col = rgb(1,0,0,0.5),
+      xlab = "Gestational Age",
+      main = "Histogram of Gestational Age")
>
> lines(density(Data2.smoke$gestation),col = "red", lwd = 2)
>
> hist(Data2.nonsmoke$gestation,
+      breaks = 100,
+      probability = TRUE,
+      col = rgb(0,0,1,0.5),
+      add = T)
>
> lines(density(Data2.nonsmoke$gestation),col = "blue",lwd = 2)
> legend("topright", c("smoke","nonsmoke"), fill = c("red","blue"))
>
>
>
> #boxplot
> boxplot(gestation~smoke,Data2.both,
+         xlab = "smoke level",
+         ylab = "gestational age",
+         main = "Boxplot of Gestational Age for Different Smoke Level",
+         cex.main = 1,
+         col = c("darkred","#E69F00","#56B4E9","yellow"))
>
>
```

```
>
> #frenquency of different weeks
> Data2.smoke.35weeks <- subset(Data2.smoke,
+                               Data2.smoke$gestation < 245)
> nrow(Data2.smoke.35weeks)
[1] 14
> Data2.smoke.36weeks <- subset(Data2.smoke,
+                               Data2.smoke$gestation < 252)
> nrow(Data2.smoke.36weeks)
[1] 25
> Data2.smoke.37weeks <- subset(Data2.smoke,
+                               Data2.smoke$gestation < 259)
> nrow(Data2.smoke.37weeks)
[1] 41
>
> Data2.smoke.38weeks <- subset(Data2.smoke,
+                               Data2.smoke$gestation < 266)
> nrow(Data2.smoke.38weeks)
[1] 66
> Data2.smoke.39weeks <- subset(Data2.smoke,
+                               Data2.smoke$gestation < 273)
> nrow(Data2.smoke.39weeks)
[1] 141
>
> #non smoker
> Data2.nonsmoke.35weeks <- subset(Data2.nonsmoke,
+                               Data2.nonsmoke$gestation < 245)
> nrow(Data2.nonsmoke.35weeks)
[1] 19
> Data2.nonsmoke.36weeks <- subset(Data2.nonsmoke,
+                               Data2.nonsmoke$gestation < 252)
> nrow(Data2.nonsmoke.36weeks)
[1] 34
> Data2.nonsmoke.37weeks <- subset(Data2.nonsmoke,
+                               Data2.nonsmoke$gestation < 259)
> nrow(Data2.nonsmoke.37weeks)
[1] 56
>
> Data2.nonsmoke.38weeks <- subset(Data2.nonsmoke,
+                               Data2.nonsmoke$gestation < 266)
> nrow(Data2.nonsmoke.38weeks)
```

```
[1] 86
> Data2.nonsmoke.39weeks <- subset(Data2.nonsmoke,
+                                Data2.nonsmoke$gestation < 273)
> nrow(Data2.nonsmoke.39weeks)
[1] 168
>
> #qqplot
> qqnorm(Data2.smoke$gestation, main = "Normal Q-Q Plot of smoking
group")
> qqline(Data2.smoke$gestation)
> qqnorm(Data2.nonsmoke$gestation,
+        main = "Normal Q-Q Plot of Nonsmoking Group")
> qqline(Data2.nonsmoke$gestation)
> qqplot(Data2.smoke$gestation, Data2.nonsmoke$gestation,
+        main = "Quantile-Quantile Plot Comparison for two distribution",
+        font.main = 8, cex.main = 1,
+        xlab = "smoking group",
+        ylab = "nonsmoking group")
> abline (c(0,1))
>
> #caculation of t-test
> mean(Data2.nonsmoke$gestation)-mean(Data2.smoke$gestation)
[1] 2.173941
> sqrt(((sd(Data2.nonsmoke$gestation))^2/nrow(Data2.nonsmoke))+
+        ((sd(Data2.smoke$gestation))^2/nrow(Data2.smoke)))
[1] 0.9255124
> t = 2.173941/0.9255124
>
>
>
>
> #calculate the kurtosis and skewness
> kurtosis(Data2.smoke$gestation)
[1] 5.083699
> kurtosis(Data2.nonsmoke$gestation)
[1] 11.73105
> skewness(Data2.smoke$gestation)
[1] -0.2293753
> skewness(Data2.nonsmoke$gestation)
[1] -1.071203
```