In [6]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import keras
pd.set_option('display.float_format', lambda x: '%.4f' % x)
import seaborn as sns
sns.set_context("paper", font_scale=1.3)
sns.set_style('white')
import warnings
warnings.filterwarnings('ignore')
from time import time
import matplotlib.ticker as tkr
from scipy import stats
from statsmodels.tsa.stattools import adfuller
from sklearn import preprocessing
from statsmodels.tsa.stattools import pacf
%matplotlib inline
import math
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import *
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from keras.callbacks import EarlyStopping
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-6-0ca124557f1f> in <module>
      2 import matplotlib.pyplot as plt
      3 import pandas as pd
----> 4 import keras
      5 pd.set_option('display.float_format', lambda x: '%.4f' % x)
      6 import seaborn as sns

ModuleNotFoundError: No module named 'keras'
```

In [9]:
```
df=pd.read_csv("C:/Users/minimini/OneDrive/MSc.DS/ML/LungCapData.csv")
```

In [10]:
```
df
```

Out[10]:

	LungCap	Age	Height	Smoke	Gender	Caesarean
0	6.4750	6	62.1000	no	male	no
1	10.1250	18	74.7000	yes	female	no
2	9.5500	16	69.7000	no	female	yes
3	11.1250	14	71.0000	no	male	no
4	4.8000	5	56.9000	no	male	no
5	6.2250	11	58.7000	no	female	no
6	4.9500	8	63.3000	no	male	yes
7	7.3250	11	70.4000	no	male	no
8	8.8750	15	70.5000	no	male	no
9	6.8000	11	59.2000	no	male	no
10	11.5000	19	76.4000	no	male	yes
11	10.9250	17	71.7000	no	male	no
12	6.5250	12	57.5000	no	male	no
13	6.0000	10	61.1000	no	female	no
14	7.8250	10	61.2000	no	male	no
15	9.5250	13	63.5000	no	male	yes
16	7.8750	15	59.2000	no	male	no
17	5.0500	8	56.1000	no	male	no
18	7.0250	11	61.2000	yes	female	no
19	9.5250	14	70.6000	no	female	no
20	3.9750	6	57.3000	no	male	no
21	5.3250	8	59.7000	no	female	no
22	10.0250	16	72.4000	no	male	no
23	8.7250	11	68.0000	no	male	yes
24	9.3750	11	65.7000	no	female	no
25	8.3500	12	61.3000	no	male	yes
26	6.7500	12	60.7000	no	female	no
27	9.0250	9	65.6000	no	male	no
28	1.1250	4	48.7000	no	female	no
29	10.4750	18	72.0000	yes	female	no
...	...	...	...	...	...	...
695	8.2500	14	67.5000	no	male	yes
696	6.3750	9	59.9000	no	female	yes
697	6.4500	16	66.5000	yes	male	no

	LungCap	Age	Height	Smoke	Gender	Caesarean
698	7.9500	15	67.3000	yes	female	no
699	7.4250	13	62.7000	no	male	no
700	6.6000	10	58.4000	no	female	no
701	10.4000	14	71.9000	no	female	no
702	9.6750	12	66.0000	no	male	no
703	3.6000	7	53.9000	no	male	no
704	6.2250	15	58.9000	no	female	no
705	12.4250	19	73.6000	no	female	yes
706	9.1750	15	71.1000	no	male	no
707	5.2750	12	58.5000	no	female	yes
708	6.9000	15	64.5000	no	female	no
709	4.8500	10	63.7000	no	female	no
710	12.3250	17	73.5000	no	male	no
711	4.6250	5	55.6000	no	female	yes
712	3.4250	3	51.0000	no	male	yes
713	9.3250	14	65.5000	no	male	no
714	9.9250	16	68.3000	no	female	no
715	8.7250	19	68.4000	no	female	no
716	7.0750	11	66.7000	no	male	yes
717	8.8250	16	71.3000	yes	female	no
718	7.1750	17	68.8000	no	male	yes
719	7.3250	9	66.3000	no	male	no
720	5.7250	9	56.0000	no	female	no
721	9.0500	18	72.0000	yes	male	yes
722	3.8500	11	60.5000	yes	female	no
723	9.8250	15	64.9000	no	female	no
724	7.1000	10	67.7000	no	male	no

725 rows × 6 columns

In [11]:
```
df.head(5)
```

Out[11]:

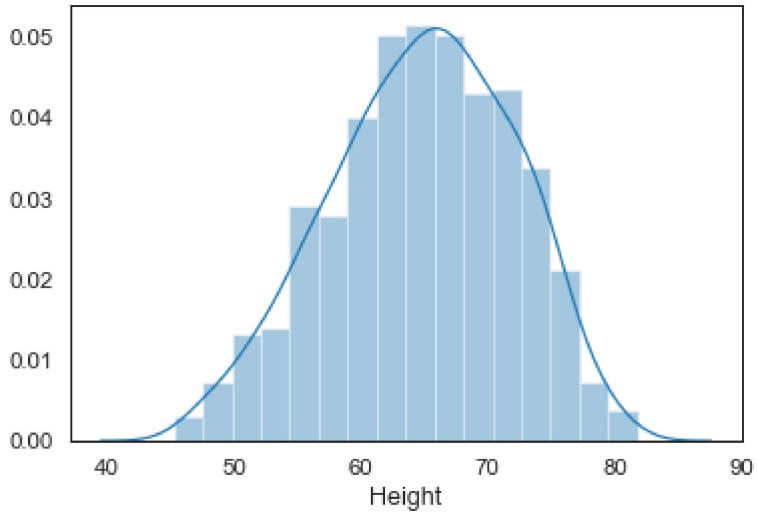
	LungCap	Age	Height	Smoke	Gender	Caesarean
0	6.4750	6	62.1000	no	male	no
1	10.1250	18	74.7000	yes	female	no
2	9.5500	16	69.7000	no	female	yes
3	11.1250	14	71.0000	no	male	no
4	4.8000	5	56.9000	no	male	no

In [14]:
```
stat, p = stats.normaltest(df.Height)
print('Statistics=%.3f, p=%.3f' % (stat, p))
alpha = 0.05
if p > alpha:
    print('Data looks Gaussian (fail to reject H0)')
else:
    print('Data does not look Gaussian (reject H0)')
```

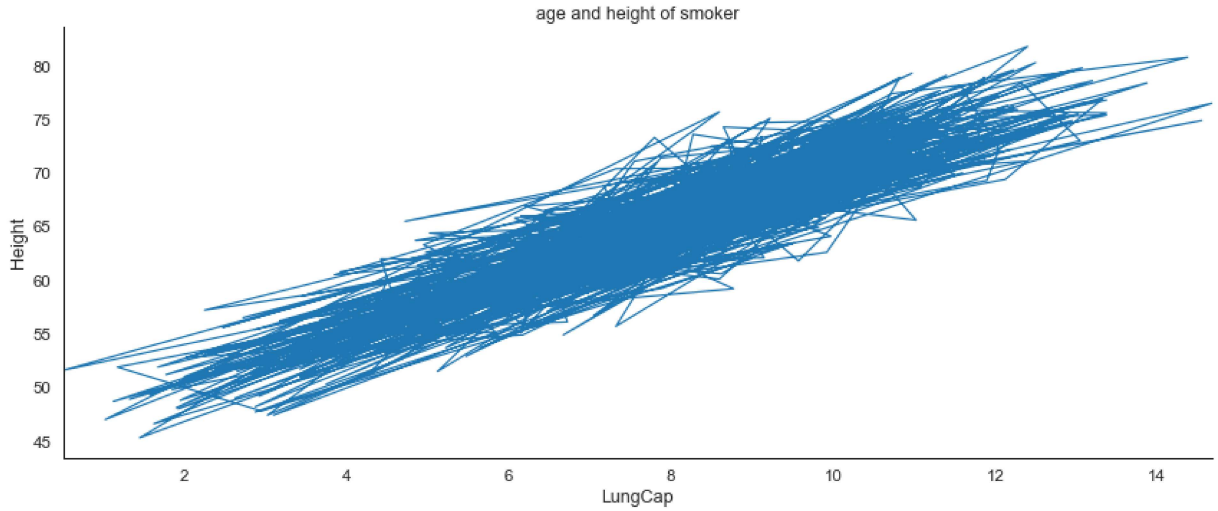
```
Statistics=19.442, p=0.000
Data does not look Gaussian (reject H0)
```

In [16]:
```
sns.distplot(df.Height);
print( 'Kurtosis of normal distribution: {}'.format(stats.kurtosis(df.Height)))
print( 'Skewness of normal distribution: {}'.format(stats.skew(df.Height)))
```

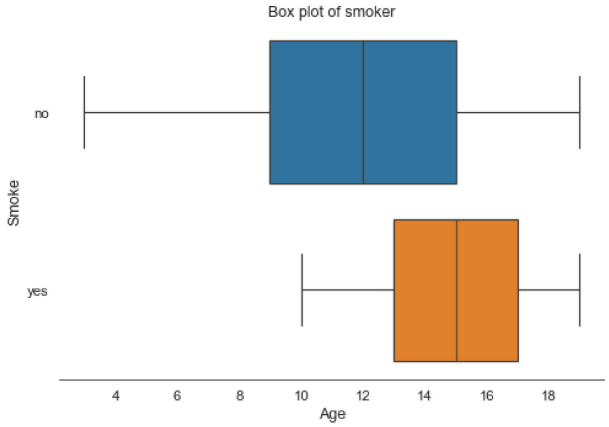
```
Kurtosis of normal distribution: -0.5044073147432249
Skewness of normal distribution: -0.22602477629651135
```

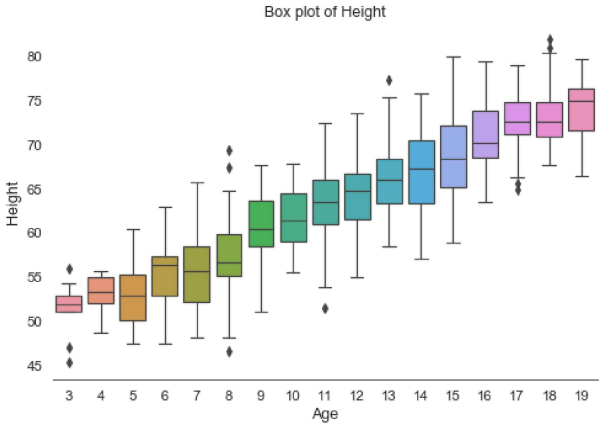


In [18]:
```
df1=df.loc[:,['LungCap','Height']]
df1.set_index('LungCap',inplace=True)
df1.plot(figsize=(12,5))
plt.ylabel('Height')
plt.legend().set_visible(False)
plt.tight_layout()
plt.title('age and height of smoker')
sns.despine(top=True)
plt.show();
```



In [19]:
```
plt.figure(figsize=(14,5))
plt.subplot(1,2,1)
plt.subplots_adjust(wspace=0.2)
sns.boxplot(x="Age", y="Smoke", data=df)
plt.xlabel('Age')
plt.title('Box plot of smoker')
sns.despine(left=True)
plt.tight_layout()
plt.subplot(1,2,2)
sns.boxplot(x="Age", y="Height", data=df)
plt.xlabel('Age')
plt.title('Box plot of Height')
sns.despine(left=True)
plt.tight_layout();
```

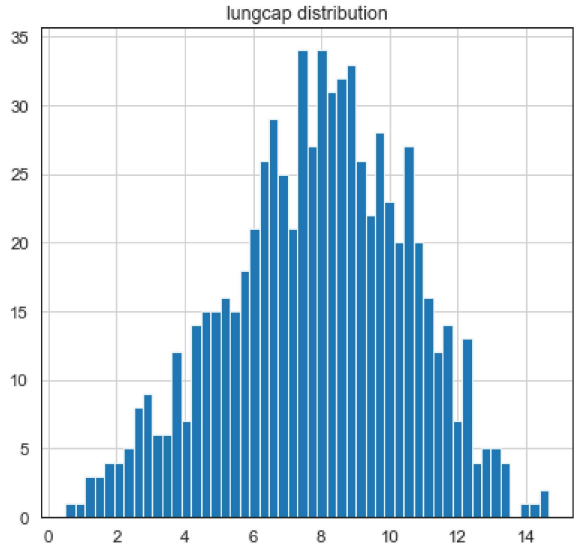


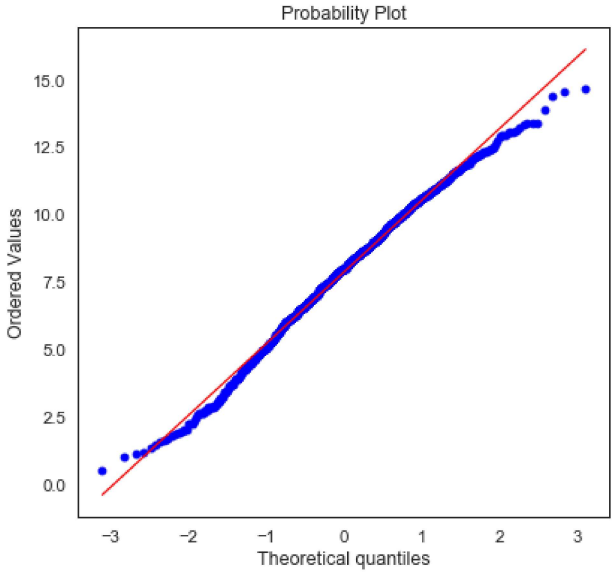


In [21]:
```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
df['LungCap'].hist(bins=50)
plt.title('lungcap distribution')
plt.subplot(1,2,2)
stats.probplot(df['LungCap'], plot=plt);
df1.describe().T
```

Out[21]:

	count	mean	std	min	25%	50%	75%	max
Height	725.0000	64.8363	7.2021	45.3000	59.9000	65.4000	70.3000	81.8000



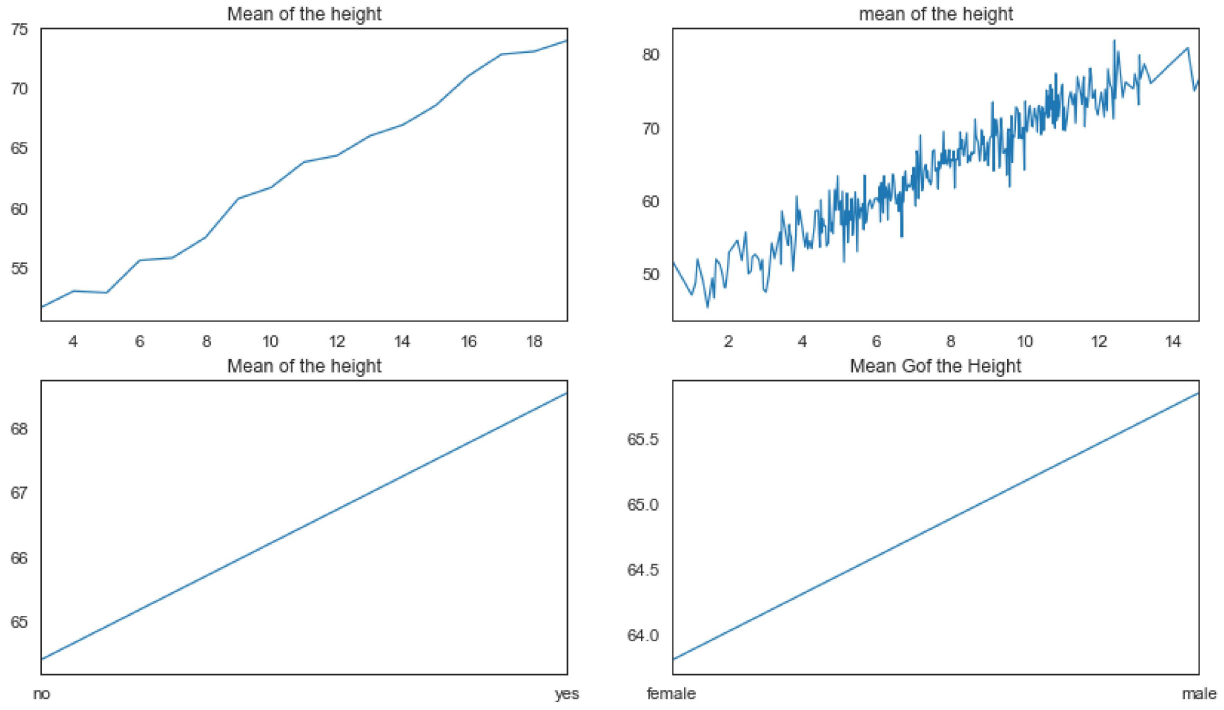


In [22]:
```
plt.figure(figsize=(14,8))
plt.subplot(2,2,1)
df.groupby('Age').Height.agg('mean').plot()
plt.xlabel('')
plt.title('Mean of the height')

plt.subplot(2,2,2)
df.groupby('LungCap').Height.agg('mean').plot()
plt.xlabel('')
plt.title('mean of the height')

plt.subplot(2,2,3)
df.groupby('Smoke').Height.agg('mean').plot()
plt.xlabel('')
plt.title('Mean of the height')

plt.subplot(2,2,4)
df.groupby('Gender').Height.agg('mean').plot()
plt.xlabel('')
plt.title('Mean Gof the Height');
```



In [ ]:
```
```