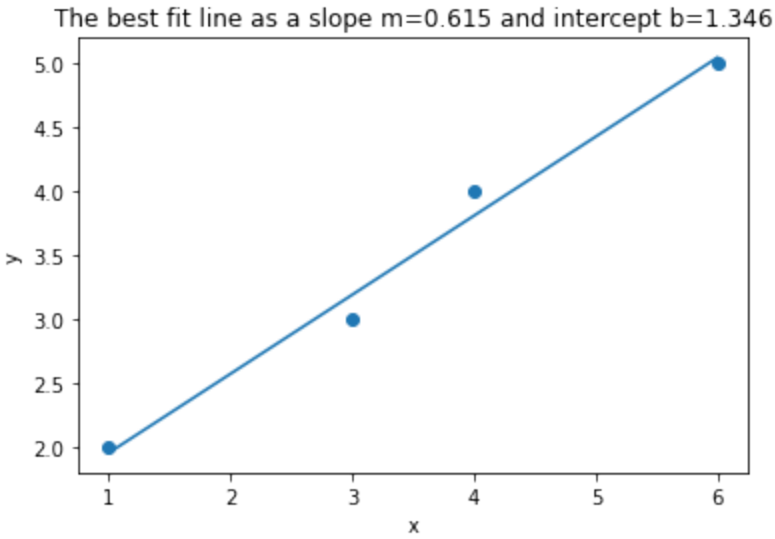
Fit a Line Answers

In [1]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

In [2]:
```
x=np.array([1.,3,4,6])
y=np.array([2,3,4,5])

xy=x*y
n=len(x)
m=(n*sum(xy)-sum(x)*sum(y))/(n*sum(x*x)-sum(x)**2)
b=(sum(x*x)*sum(y)-sum(xy)*sum(x))/(n*sum(x*x)-sum(x)**2)
x_fit=np.linspace(min(x),max(x))
y_fit=m*x_fit+b

fig,ax=plt.subplots()
ax.scatter(x,y)
ax.plot(x_fit,y_fit)
ax.set_xlabel('x')
ax.set_ylabel('y')
title='The best fit line as a slope m={:.3f} and intercept b={:.3f}'.format(m,b)
ax.set_title(title)
```

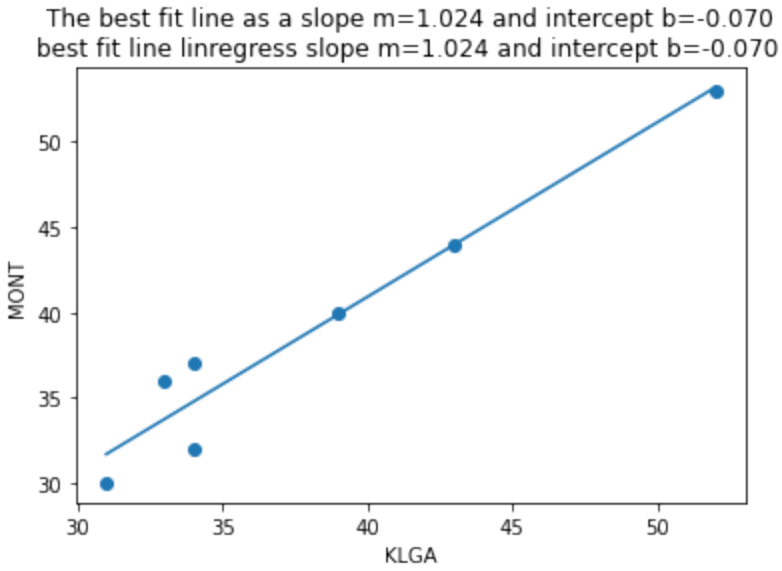
Out[2]:
```
Text(0.5, 1.0, 'The best fit line as a slope m=0.615 and intercept b=1.346')
```



In [19]:
```
KLGA=np.array([52,43,39,34,33,31,34])
MONT=np.array([53,44,40,37,36,30,32])

x=KLGA
y=MONT
xy=x*y
n=len(x)
m=(n*sum(xy)-sum(x)*sum(y))/(n*sum(x*x)-sum(x)**2)
b=(sum(x*x)*sum(y)-sum(xy)*sum(x))/(n*sum(x*x)-sum(x)**2)
x_fit=np.linspace(min(x),max(x))
y_fit=m*x_fit+b
```

```
fig,ax=plt.subplots()
ax.scatter(x,y)
ax.plot(x_fit,y_fit)
ax.set_xlabel('KLGA')
ax.set_ylabel('MONT')

linregress_out=stats.linregress(x,y)

title=('The best fit line as a slope m={:.3f} and intercept b={:.3f}'.format(m,b
       '\nbest fit line linregress slope m={:.3f} and intercept b={:.3f} '\
       .format(linregress_out[0],linregress_out[1]))
ax.set_title(title)
```

Out[19]:
```
Text(0.5, 1.0, 'The best fit line as a slope m=1.024 and intercept b=-0.070\nbes
t fit line linregress slope m=1.024 and intercept b=-0.070 ')
```

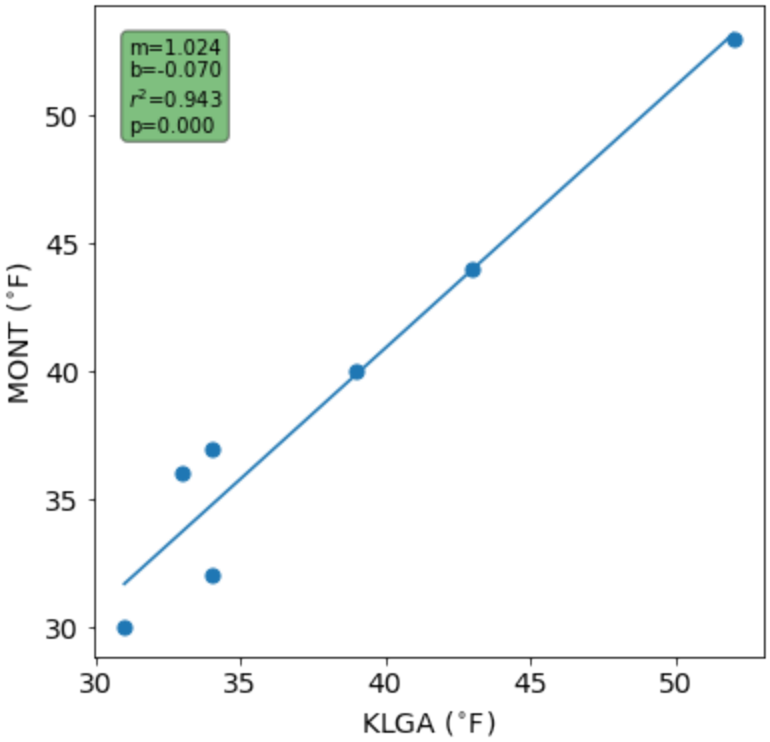


Now that is only a decent plot. We don't have units on the axis and I will show you how to represent the m,b,r^2 and p on the graph. Here we go.

In [15]:
```
KLGA=np.array([52,43,39,34,33,31,34])
MONT=np.array([53,44,40,37,36,30,32])

fig,ax=plt.subplots()
fig.set_size_inches(6,6)
ax.scatter(KLGA,MONT,s=50)

ax.set_xlabel('KLGA ($^{\circ}$F)',fontsize=14)
ax.set_ylabel('MONT ($^{\circ}$F)',fontsize=14)

ax.tick_params(axis='both', which='major', labelsize=14) #This increases the for

linregress_out=stats.linregress(KLGA,MONT)

x_fit=np.linspace(min(KLGA),max(KLGA))
y_fit=linregress_out[0]*x_fit+linregress_out[1]
ax.plot(x_fit,y_fit)

props=dict(boxstyle='round',facecolor='green',alpha=0.5)

textstr='m={:.3f}\nb={:.3f}\n$r^2$={:.3f}\np={:.3f}'\
            .format(linregress_out[0],linregress_out[1]\
                    ,linregress_out[2]**2,linregress_out[3])
```

```
ax.text(0.05,0.95,textstr,transform=ax.transAxes,fontsize=10\
        ,verticalalignment='top',bbox=props)
```

Out[15]: Text(0.05, 0.95, 'm=1.024\nb=-0.070\n$r^2$=0.943\np=0.000')



Now we are starting to look professional!!! and can make a figure caption. Figure 1. Temperature at JFK Airport (KNYC) versus temperature at Laguardia Airport (KLGA). The line is the best fit linear regression.

In [ ]:

In [ ]: