Reto: Encontrar el libro que tiene mejor puntuación

Paso 1. Entender el problema:

En el archivo books.csv se encuentra una gran lista de libros. Al principio quería resolver este ejercicio parseando el archivo con expresiones regulares y encontrar las lineas que tuvieran la mayor calificación promedio (ver archivo books_parsing.py) pero luego reflexioné que ese era un acercamiento incorrecto, ya que no estaba considerando el recuento de calificaciones y el recuento de revisiones del texto.

Despues de analizar por unos minutos esto, llegué a la conclusión que el libro con mayor valoración puede ser definido de dos maneras / acercamientos: I) Aquel que tenga una calificación promedio (promedio de promedios) más una desviación estandar a la derecha y que tenga un recuento de calificaciones promedio más una desviación estandar hacia arriba, y II) Aquel que tenga una calificación promedio (promedio de promedios) más una desviación estandar a la derecha y que tenga un recuento de revisiones/reseñas de texto promedio más una desviación estandar hacia arriba

Paso 2. Tratamiento de datos.

In [28]:
```
import scipy.stats
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

In [29]:
```
df =pd.read_csv('books.csv') #Nota importante: Hay qu verificar que el archivo esté "cuadrado"
```

Durante el tratamiento de los datos, encontré que el separador de filas era una coma, lo que ocacionaba algunos problemas al momento de leer el dataframe. Con ayuda de Jupyter Notebook encontré que el problema estaba en que alhgunos libros separaban autores con comas y eso ocacionaba que existieran más columnas de las que deberían haber en el archivo.

In [30]:
```
df.columns    #Muestra las columnas del archivo CSV
```

Out[30]:
```
Index(['bookID', 'title', 'authors', 'average_rating', 'isbn', 'isbn13',
       'language_code', '  num_pages', 'ratings_count', 'text_reviews_count',
       'publication_date', 'publisher'],
      dtype='object')
```

Paso 3. Expresion de datos y parámetros estadísticos (Acercamiento I)

In [31]:
```
x = df['average_rating']      # La variable independiente es la calificació
n promedio
y = df['ratings_count']    # la variable dependiente es el recuento de cali
ficaciones
```

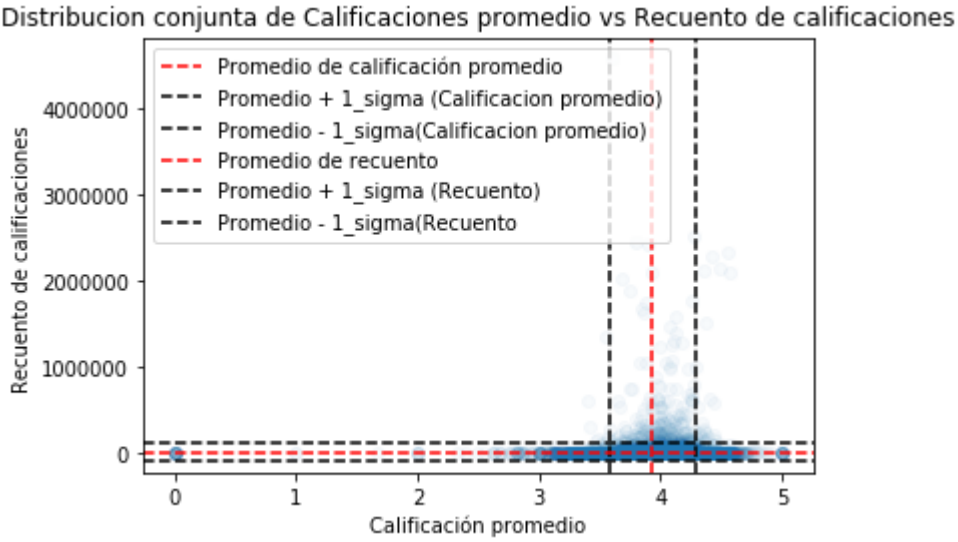
In [32]:
```
fig,ax = plt.subplots()    #Las variables de toda la vida antes de realizar
cualquier gráfica.
ax.scatter(df['average_rating'], df['ratings_count'], alpha = 0.03)
ax.set_title('Distribucion conjunta de Calificaciones promedio vs Recuento
 de calificaciones')
ax.set_xlabel('Calificación promedio')    #Título del eje x
ax.set_ylabel('Recuento de calificaciones ')    #Título del eje y

plt.axvline(np.mean(x), c = 'r', linestyle = '--', label = 'Promedio de cal
ificación promedio')
plt.axvline(np.mean(x) + np.std(x), c = 'k', linestyle = '--', label = 'Pro
medio + 1_sigma (Calificacion promedio)')
plt.axvline(np.mean(x) - np.std(x), c = 'k', linestyle = '--', label = 'Pro
medio - 1_sigma(Calificacion promedio)')

plt.axhline(np.mean(y), c = 'r', linestyle = '--', label = 'Promedio de rec
uento')
plt.axhline(np.mean(y) + np.std(y), c = 'k', linestyle = '--', label = 'Pro
medio + 1_sigma (Recuento)')
plt.axhline(np.mean(y) - np.std(y), c = 'k', linestyle = '--', label = 'Pro
medio - 1_sigma(Recuento')

ax.legend ()
```

Out[32]:
```
<matplotlib.legend.Legend at 0x26895b1ad08>
```



In [33]:
```
#La mayor calificación que está al limite de 1 desviación estandar a la der
echa / arriba.
print (f'Mean of mean rating + 1_Sigma = {np.mean(x) + np.std(x)}')
print (f'Mean of mean ratings count + 1_Sigma = {np.mean(y) + np.std(y)}')
```

```
Mean of mean rating + 1_Sigma = 4.286060002705698
Mean of mean ratings count + 1_Sigma = 130410.79608301754
```

Paso 4. Expresion de datos y parámetros estadísticos (Acercamiento II)

In [34]:
```
x = df['average_rating']      # La variable independiente es la calificació
n promedio
y = df['text_reviews_count']    # la variable dependiente es el recuento de
reseñas del texto.
```

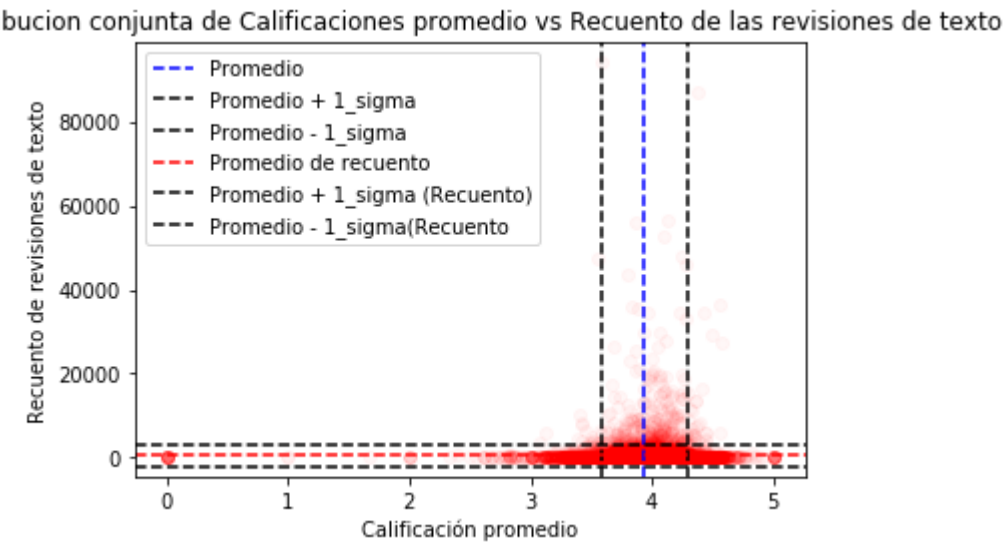
In [35]:
```
fig,ax = plt.subplots()    #Las variables de toda la vida antes de realizar cualquier gráfica.
ax.scatter(df['average_rating'], df['text_reviews_count'], c = 'red' ,alpha = 0.03)
ax.set_title('Distribucion conjunta de Calificaciones promedio vs Recuento de las revisiones de texto')
ax.set_xlabel('Calificación promedio')    #Título del eje x
ax.set_ylabel('Recuento de revisiones de texto')    #Título del eje y

plt.axvline(np.mean(x), c = 'b', linestyle = '--', label = 'Promedio')    # Dibuja una linea vertical, de color rojo,
plt.axvline(np.mean(x) + np.std(x), c = 'k', linestyle = '--', label = 'Promedio + 1_sigma')
plt.axvline(np.mean(x) - np.std(x), c = 'k', linestyle = '--', label = 'Promedio - 1_sigma')

plt.axhline(np.mean(y), c = 'r', linestyle = '--', label = 'Promedio de recuento')
plt.axhline(np.mean(y) + np.std(y), c = 'k', linestyle = '--', label = 'Promedio + 1_sigma (Recuento)')
plt.axhline(np.mean(y) - np.std(y), c = 'k', linestyle = '--', label = 'Promedio - 1_sigma(Recuento')

ax.legend ()
```

Out[35]:
```
<matplotlib.legend.Legend at 0x268959bef48>
```



In [36]:
```
#La mayor calificación que está al limite de 1 desviación estandar a la derecha / arriba.
print (f'Mean of mean rating + 1_Sigma = {np.mean(x) + np.std(x)}')
print (f'Mean of mean ratings count + 1_Sigma = {np.mean(y) + np.std(y)}')
```

```
Mean of mean rating + 1_Sigma = 4.286060002705698
Mean of mean ratings count + 1_Sigma = 3117.915340961821
```

Paso 5. interpretación de los resultados

Según los resultados del Approach I, el libro más valorado será aquel que tenga una calificación promedio cercana a 4.28 y con un conteo de calificaciones cercano a 130410. Solo dos libros se acercan a esos resultados:

BookID Title

2199 Team of Rivals: The Political Genius of Abraham Lincoln Doris Kearns Goodwin (con 4.28 y 133840)
5544 Surely You're Joking Mr. Feynman!: Adventures of a Curious Character Richard P. Feynman (con 4.28 y 106526)

Según los resultados del Approach II, el libro más valorado será aquel que tenga una calificación promedio cercana a 4.28 y con un conteo de reseñas cercano a 3118. Solo dos libros se acercan a esos resultados:

BookID Title

5544 Surely You're Joking Mr. Feynman!: Adventures of a Curious Character (con 4.28 y 3685) 11557 Swan Song (con 4.28 y 2540)

Paso 6. Toma de desicion

Con base en los análisis anteriormente expuestos, a la cercanía de los limites superiores en ambos métodos y debido a que es una componente común entre los dos acercamientos. El libro con BookID: **5544**, Titulo: **Surely You're Joking Mr. Feynman!: Adventures of a Curious Character** y Autor: **Richard P. Feynman**. Es el libro con mejor puntuación dentro del archivo.