Primera Parte

December 7, 2019

```
In [18]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         from scipy.interpolate import interp1d
         from scipy.optimize import curve_fit
         import scipy.integrate as integrate
         from tqdm import trange


         %matplotlib inline
```

1 Observaciones de Steinhardt

En esta primera parte intentaré reajustar las predicciones teóricas que aparecen Section 2.1 con los nuevos datos ajustados de Section 2.1. Para ello vamos a recopilar todos los datos usados por Section 2.1 tanto para el ajuse teórico de la función de masa de halo (Section 2.1, Section 2.1) como para el tratamiento de los datos observacionales que se muestran en la imagen. >*Theoretical halo number density as a function of halo mass and redshift (Section 2.1, Section 2.1) for the most massive halos at* $4 < z < 10$ *(shown as solid lines, with red- der colors at higher redshift) compared with observational number densities of estimated halo masses corresponding to observed star- forming galaxies at similar redshifts. Halo masses are estimated using clustering (triangle), stellar masses converted to halo masses using thelow-redshift scaling ratio* $M_H/M_\star \sim 70$ *(square), and UV luminosities converted to halo masses using ratios determined by lower-redshift abundance matching (circle), as described in ğ 2, for an overall* $M_H/M\odot \sim 120L_{UV}/L_\odot$. *These methods all give self-consistent number densities that disagree with theoretical ex- pectations.*

Para ello recopilaremos los datos de los artículos citados Section 2.1, Section 2.1, Section 2.1 y Section 2.1, donde aplicaremos la distribución de los datos encontrados en Section 2.1 para recalcular los datos obtenidos de dichas observaciones. El esquema de este Notebook será el siguiente:

1. Construiremos las distribuciones de los datos estudiados en Section 2.1.
2. Estudiaremos las predicciones Teóricas sacadas de Section 2.1 y Section 2.1.
3. Recopilaremos las observaciones de Section 2.1, Section 2.1, Section 2.1 y Section 2.1.
4. Aplicaremos las distribuciones de los datos de Section 2.1 a las funciones de masa de halo teóricas y a las derivadas por las observaciones.

1.1 Observaciones

1.1.1 Section 2.1

Los datos de Section 2.1 fueron tomados del estudio CFHTLS - *Deep Survey* analizando los saltos en las magnitudes del filtro u,g y r para calcular su redshift fotométrico. Dichos datos los podemos encontrar en el archivo 'data_cfhtls.csv'.

```
In [158]: cfhtl_data = pd.read_csv("CFHTLS_DeepSurvey/data_cfhtls.csv")
```

Como encontramos objetos donde no tenemos muestra en las magnitudes que desemos dichas observaciones las descartamos, reemplazando el hueco en blanco observado por un 'NaN'

```
In [4]: valor_a_descartar = np.sort(cfhtl_data.gmag.unique())[0]

In [5]: cfhtls_u = np.zeros(cfhtl_data.shape[0])
        cfhtls_g = np.zeros(cfhtl_data.shape[0])
        cfhtls_r = np.zeros(cfhtl_data.shape[0])
        cfhtls_i = np.zeros(cfhtl_data.shape[0])
        cfhtls_z = np.zeros(cfhtl_data.shape[0])

        for k in range(cfhtl_data.shape[0]):
            u = cfhtl_data.umag.values[k]
            g = cfhtl_data.gmag.values[k]
            r = cfhtl_data.rmag.values[k]
            i = cfhtl_data.imag.values[k]
            z = cfhtl_data.zmag.values[k]
            cfhtls_u[k] = np.float(u.replace(valor_a_descartar,"NaN"))
            cfhtls_g[k] = np.float(g.replace(valor_a_descartar,"NaN"))
            cfhtls_r[k] = np.float(r.replace(valor_a_descartar,"NaN"))
            cfhtls_i[k] = np.float(i.replace(valor_a_descartar,"NaN"))
            cfhtls_z[k] = np.float(z.replace(valor_a_descartar,"NaN"))

In [6]: mask = np.isnan(cfhtls_u) * np.isnan(cfhtls_g) * np.isnan(cfhtls_r) * np.isnan(cfhtls_
        cfhtls_u = cfhtls_u[~mask]
        cfhtls_g = cfhtls_g[~mask]
        cfhtls_r = cfhtls_r[~mask]
        cfhtls_i = cfhtls_i[~mask]
        cfhtls_z = cfhtls_z[~mask]
```

Una vez limpiado la muestra vacía gráficamos los resultados para detectar los saltos en las distintas magnitudes > Saltos en u:

$$1.5 < (ug)1.0 < (gr) < 1.21.5(gr) < (ug)0.7$$

Saltos en g:

$$1.0 < (gr)1.0 < (ri) < 1.01.5(ri) < (gr)0.8$$

Saltos en z:

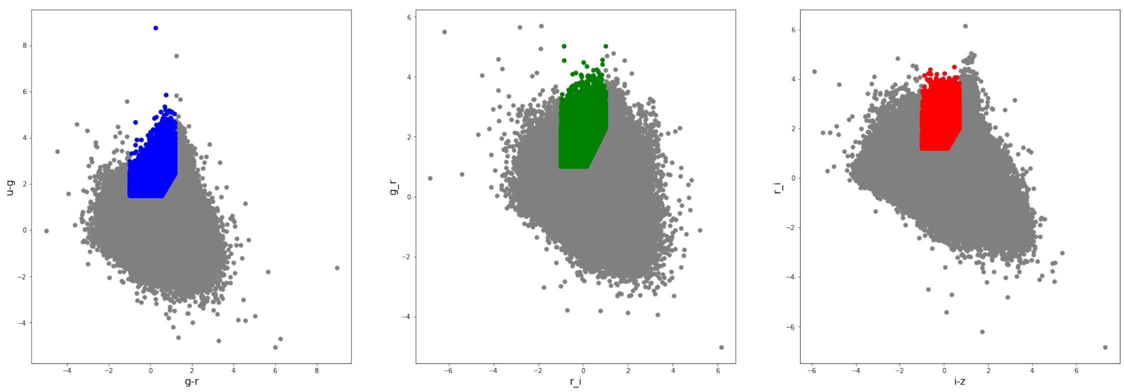
$$1.2 < (ri)1.0 < (iz) < 0.71.5(iz) < (ri)1.0$$

```
In [7]: u_g = cfhtls_u - cfhtls_g
        g_r = cfhtls_g - cfhtls_r
        r_i = cfhtls_r - cfhtls_i
        i_z = cfhtls_i - cfhtls_z

        u_drop= (u_g > 1.5) & (g_r > -1) & (g_r < 1.2) & ((u_g - 0.7)>1.5*g_r)
        g_drop= (g_r > 1.0) & (r_i > -1) & (r_i < 1.0) & ((g_r - 0.8)>1.5*r_i)
        z_drop= (r_i > 1.2) & (i_z > -1) & (i_z < 0.7) & ((r_i - 1.0)>1.5*i_z)
```

```
/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:6: RuntimeWarning: inv

/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:6: RuntimeWarning: inv

/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7: RuntimeWarning: inv
  import sys
/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7: RuntimeWarning: inv
  import sys
/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:8: RuntimeWarning: inv

/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:8: RuntimeWarning: inv
```

Y realizamos las gráficas de los CFHTLS - Deep Survey

```
In [86]: plt.figure(figsize=(30,10))
         plt.title("Hildergberg CFHTLS Deep Survey", fontsize=20)

         ax1=plt.subplot(1,3,1)
         ax1.scatter(g_r,u_g, color="grey")
         ax1.scatter(g_r[u_drop],u_g[u_drop], color="blue")
         ax1.set_xlabel("g-r", fontsize=16)
         ax1.set_ylabel("u-g", fontsize=16)

         ax2=plt.subplot(1,3,2)
         ax2.scatter(r_i,g_r, color="grey")
         ax2.scatter(r_i[g_drop],g_r[g_drop], color="green")
         ax2.set_xlabel("r-i", fontsize=16)
         ax2.set_ylabel("g-r", fontsize=16)

         ax3=plt.subplot(1,3,3)
         ax3.scatter(i_z,r_i, color="grey")
         ax3.scatter(i_z[z_drop],r_i[z_drop], color="red")
         ax3.set_xlabel("i-z", fontsize=16)
         ax3.set_ylabel("r-i", fontsize=16)

         plt.savefig("Test.png")
```



1.1.2 Sec 3 -- The LBG samples

Las simulaciones toman como referencia para la creación del catálog el número de muestras registradas en la banda *i* de las observaciones

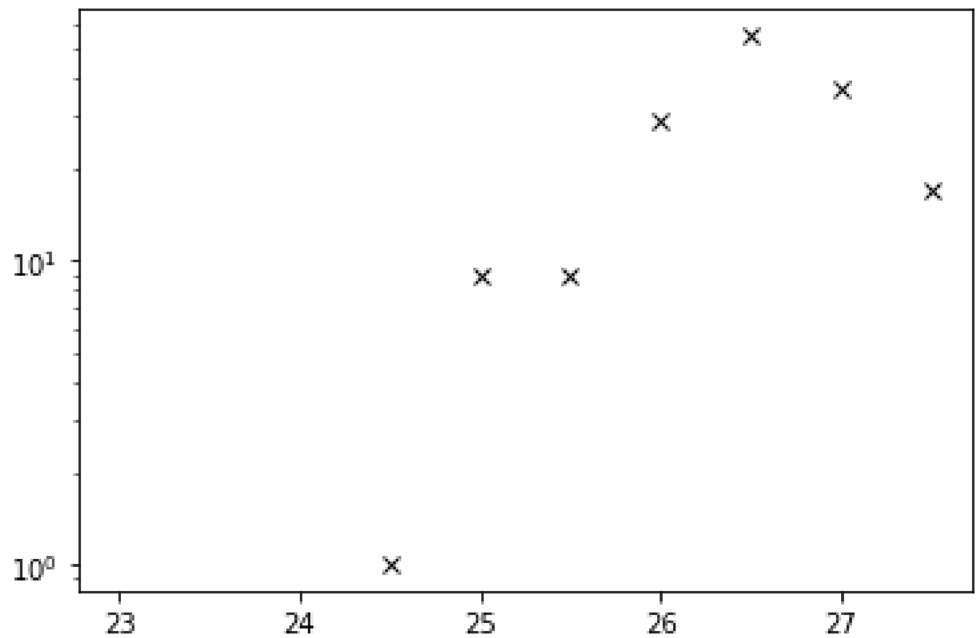
```
In [8]: i_counts = np.count_nonzero(cfhtls_i)
        print("El número de cuentas registradas en la banda i es de %.i" % (i_counts))
```

```
El número de cuentas registradas en la banda i es de 2254580
```

Sigamos todos los pasos que sigue HILDEBRANDT

```
In [34]: i_udrops=cfhtls_i[g_drop]
         i_bins = np.arange(23,28,0.5)

         i_counts = np.array([np.count_nonzero(i_udrops==i) for i in i_bins])
```

```
In [37]: plt.plot(i_bins,i_counts,"kx")
         plt.yscale("log")
```



1.2 Método de enlace de abundancias

El método de enlace de abundancias que se muestra en Section 2.1 se base en enlazar la función de luminosidad con la función de masa de halo para así obtener la masa del halo en función de la luminosidad de la galaxia. La hipótesis aceptada para usar este método es la suposición de que las galaxias más masivas se encuentran alojadas en los halos más masivos, de esta manera enlazan la función de luminosidad de Shecter cuyo ajuste es estudiado en Section 2.1 con la función de masa de halo obtenida de las simulaciones de Bolshoi por medio del número de densidad obtenido por ambas funciones, como se puede observar en la figura de más abajo

1.2.1 Datos de Section 2.1

Los datos de Section 2.1 es una submuestra de las casi 7500 galaxias estudiadas en Section 2.1 en donde da una aproximación de la función de luminosidad de Shecter para diferentes rangos de redshift de la muestra estudiada:

$$\varphi(M) = 0.4\ln(10)\varphi^* 10^{-0.4(M-M^*)(\alpha+1)} e^{-10^{-0.4(M-M^*)}}$$

En donde los ajustes obtenidos para los valores de α, M^*, φ^* son:

Redshift	M^*	α	φ^* $(\mathrm{Mpc}^* - 3)$
4	$-20.73^{+0.09}_{-0.09}$	$-1.56^{+0.06}_{-0.05}$	$\left(14.1^{+2.05}_{-1.85}\right) \times 10^{-4}$
5	$-20.81^{+0.13}_{-0.12}$	$-1.67^{+0.05}_{-0.06}$	$\left(8.95^{+1.92}_{-1.31}\right) \times 10^{-4}$
6	$-21.13^{+0.25}_{-0.31}$	$-2.02^{+0.10}_{-0.10}$	$\left(1.86^{+0.94}_{-0.8}\right) \times 10^{-4}$
7	$-21.03^{+0.37}_{-0.50}$	$-2.03^{+0.21}_{-0.20}$	$\left(1.57^{+1.49}_{-0.95}\right) \times 10^{-4}$
8	$-20.89^{+0.74}_{-1.08}$	$-2.36^{+0.54}_{-0.40}$	$\left(0.72^{+2.52}_{-0.65}\right) \times 10^{-4}$

```
In [2]: def phi(M,z):

            m,a,p = [0,0,0]
            if z>3.5 and z<= 4.5:
                m,a,p = [-20.73,-1.56,14.1*10**(-4)]
            elif z>4.5 and z<=5.5:
                m,a,p = [-20.81,-1.67,8.95*10**(-4)]
            elif z>5.5 and z<=6.5:
                m,a,p = [-21.13,-2.02,1.86*10**(-4)]
            elif z>6.5 and z<=7.5:
                m,a,p = [-21.03,-2.03,1.57*10**(-4)]
            elif z>7.5 and z<=8.5:
                m,a,p = [-20.89,-2.36,0.72*10**(-4)]


            part1=0.4*np.log(10)
            paw1= -0.4*(M-m)*(a+1)
            paw2= -np.power(10,-0.4*(M-m))
            part2=np.power(10,paw1)
            part3=np.exp(paw2)

            phi_value = part1*part2*part3*p

            return phi_value
```

Intentemos replicar los datos obtenidos en los estudios de Finkelstein y veamos como puede encajar las nuevas métricas obtenidos por Section 2.1 y las nuevas observaciones de Section 2.1

```
In [3]: fink_a = pd.read_csv("Finkelstein/ApJ810_2015/apj810.tsv", sep=";", skiprows=53, index_
        fink_b = pd.read_csv("Finkelstein/ApJ814_2015/apj814.tsv", sep=";", skiprows=72, index_

        fink_a = fink_a.iloc[2:,:]
        fink_b = fink_b.iloc[2:,:]
```

```
In [4]: fink_b.columns
```

```python
col_fb= ['_RAJ2000', '_DEJ2000', 'ID', 'RAJ2000', 'DEJ2000', 'z', 'b_z', 'B_z',
       'logM*', 'b_logM*', 'B_logM*', 'Age', 'b_Age', 'B_Age', 'E(B-V)',
       'b_E(B-V)', 'B_E(B-V)', 'SFR', 'b_SFR', 'B_SFR', 'Simbad']

ids_fb= fink_b.ID.values
fink_a.ID = np.array([i.replace(" ","") for i in fink_a.ID.values])
test=fink_a.ID.isin(ids_fb)
sample = fink_a[test]
not_sample = fink_b[~(fink_b.ID.isin(fink_a.ID.values))]
sample
```

Out[4]:

```
          _RAJ2000      _DEJ2000            ID                     HRG14  \
30    053.09684000  -27.86607389  z4_GSD_34736  HRG14 J033223.24-275157.9
47    053.08689111  -27.84413889  z4_GSD_30292  HRG14 J033220.85-275038.9
177   053.08736806  -27.83953500  z4_GSD_29028  HRG14 J033220.97-275022.3
211   053.12141389  -27.81462111  z4_GSD_21002  HRG14 J033229.14-274852.6
215   053.03123889  -27.78521500  z4_GSD_11269  HRG14 J033207.50-274706.8
...            ...           ...           ...                       ...
7261  189.35688806  +62.29531889  z7_GNW_24443  HRG14 J123725.65+621743.1
7267  189.03248611  +62.21641500  z7_GNW_17001  HRG14 J123607.80+621259.1
7269  189.36171000  +62.29437306  z7_GNW_24671  HRG14 J123726.81+621739.7
7278  189.27339194  +62.32478306  z7_GNW_19939  HRG14 J123705.61+621929.2
7344  053.28171194  -27.86769889  z7_PAR2_3098  HRG14 J033307.61-275203.7

         RAJ2000     DEJ2000  zphot b_zphot B_zphot 1500Mag b_1500Mag  \
30    053.096840  -27.866074   3.51    0.38    3.64  -21.36    -21.50
47    053.086891  -27.844139   3.54    0.40    3.70  -21.03    -21.23
177   053.087368  -27.839535   3.60    3.48    3.73  -21.03    -21.14
211   053.121414  -27.814621   3.63    3.57    3.73  -21.18    -21.27
215   053.031239  -27.785215   3.63    3.53    3.77  -21.30    -21.41
...          ...         ...    ...     ...     ...     ...       ...
7261  189.356888  +62.295319   6.66    6.48    6.77  -21.83    -21.92
7267  189.032486  +62.216415   6.84    6.58    7.03  -21.12    -21.21
7269  189.361710  +62.294373   6.93    6.27    7.38  -21.22    -21.45
7278  189.273392  +62.324783   7.24    6.81    7.69  -21.04    -21.23
7344  053.281712  -27.867699   6.79    6.57    7.53  -21.04    -21.44

     B_1500Mag
30      -21.32
47      -20.98
177     -20.96
211     -21.13
215     -21.20
...        ...
7261    -21.73
7267    -20.97
7269    -20.92
7278    -20.79
```

```
7344    -20.93

[118 rows x 12 columns]
```

```
In [5]: m_fa = fink_a.loc[:,["zphot","1500Mag"]].astype(float)
        z_fa = fink_a.zphot.astype(float)

        mask_z4 = (z_fa >3.5) & (z_fa <= 4.5)
        mask_z5 = (z_fa >4.5) & (z_fa <= 5.5)
        mask_z6 = (z_fa >5.5) & (z_fa <= 6.5)
        mask_z7 = (z_fa >6.5) & (z_fa <= 7.5)
        mask_z8 = (z_fa >7.5) & (z_fa <= 8.5)

        m_z4=m_fa[mask_z4]
        m_z5=m_fa[mask_z5]
        m_z6=m_fa[mask_z6]
        m_z7=m_fa[mask_z7]
        m_z8=m_fa[mask_z8]
```

```
In [6]: def phi_obs(M,m_z, delta_M):
            n= M.shape[0]
            p_values=np.zeros(n)
            j=0
            for i in M:
                mask = (m_z["1500Mag"].values >i-delta_M) & (m_z["1500Mag"].values < i+delta_M
                p_values[j] = np.sum(mask)
                j+=1

            return p_values/np.count_nonzero(m_z["1500Mag"].values)
```

```
In [7]: phi_fink_z4=pd.read_csv("umachine-dr1/observational_constraints/finkelstein_z3.5_z4.5.
                                skiprows=4, sep="\s", names=["bMag","BMag","phi","bphi","Bphi"]

        phi_fink_z5=pd.read_csv("umachine-dr1/observational_constraints/finkelstein_z4.5_z5.5.
                                skiprows=4, sep="\s", names=["bMag","BMag","phi","bphi","Bphi"]

        phi_fink_z6=pd.read_csv("umachine-dr1/observational_constraints/finkelstein_z5.5_z6.5.
                                skiprows=4, sep="\s", names=["bMag","BMag","phi","bphi","Bphi"]

        phi_fink_z7=pd.read_csv("umachine-dr1/observational_constraints/finkelstein_z6.5_z7.5.
                                skiprows=4, sep="\s", names=["bMag","BMag","phi","bphi","Bphi"]

        phi_fink_z8=pd.read_csv("umachine-dr1/observational_constraints/finkelstein_z7.5_z8.5.
                                skiprows=4, sep="\s", names=["bMag","BMag","phi","bphi","Bphi"]
```

```
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche

C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  """
```

```
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche

C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  # This is added back by InteractiveShellApp.init_path()
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
```

In [8]:
```
M=np.arange(-23,-16.9,0.1)
```

In [9]:
```
hmf_z4=pd.read_csv("umachine-dr1/data/hmfs/hmf_a0.202435.dat", skiprows=2,sep="\s")
uvlf_z4=pd.read_csv("umachine-dr1/data/uvlfs/uvlf_a0.202435.dat", skiprows=3,sep="\s")

hmf_z5=pd.read_csv("umachine-dr1/data/hmfs/hmf_a0.166998.dat", skiprows=2,sep="\s")
uvlf_z5=pd.read_csv("umachine-dr1/data/uvlfs/uvlf_a0.166998.dat", skiprows=3,sep="\s")

hmf_z6=pd.read_csv("umachine-dr1/data/hmfs/hmf_a0.141685.dat", skiprows=2,sep="\s")
uvlf_z6=pd.read_csv("umachine-dr1/data/uvlfs/uvlf_a0.141685.dat", skiprows=3,sep="\s")

hmf_z7=pd.read_csv("umachine-dr1/data/hmfs/hmf_a0.126498.dat", skiprows=2,sep="\s")
uvlf_z7=pd.read_csv("umachine-dr1/data/uvlfs/uvlf_a0.126498.dat", skiprows=3,sep="\s")

hmf_z8=pd.read_csv("umachine-dr1/data/hmfs/hmf_a0.112998.dat", skiprows=2,sep="\s")
uvlf_z8=pd.read_csv("umachine-dr1/data/uvlfs/uvlf_a0.112998.dat", skiprows=3,sep="\s")


hmfs_list=[hmf_z4,hmf_z5,hmf_z6,hmf_z7,hmf_z8]
uvlfs_list=[uvlf_z4,uvlf_z5,uvlf_z6,uvlf_z7,uvlf_z8]
```

```
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  """Entry point for launching an IPython kernel.
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche

C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  after removing the cwd from sys.path.
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  """
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  import sys
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche

C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  # Remove the CWD from sys.path while we load stuff.
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  # This is added back by InteractiveShellApp.init_path()
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  del sys.path[0]
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
```

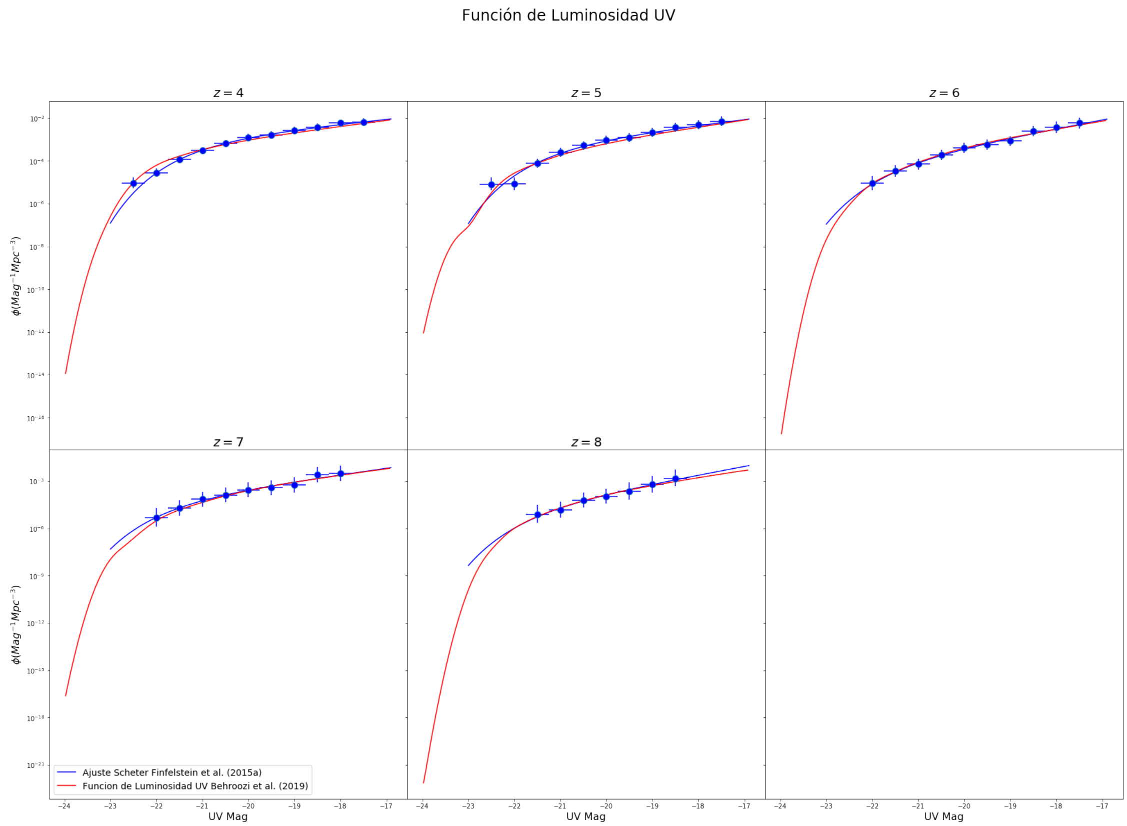
In [10]:
```
yerror_low=np.power(10,phi_fink_z4.phi.values)- np.power(10,phi_fink_z4.phi.values-phi
yerror_up=(np.power(10,phi_fink_z4.phi.values + phi_fink_z4.Bphi.values)- np.power(10
yerror_z4=np.array((yerror_low,yerror_up))
xerror_z4=(phi_fink_z4.BMag.values - phi_fink_z4.bMag.values)/2

yerror_low=np.power(10,phi_fink_z5.phi.values)- np.power(10,phi_fink_z5.phi.values-phi
yerror_up=(np.power(10,phi_fink_z5.phi.values + phi_fink_z5.Bphi.values)- np.power(10
yerror_z5=np.array((yerror_low,yerror_up))
xerror_z5=(phi_fink_z5.BMag.values - phi_fink_z5.bMag.values)/2

yerror_low=np.power(10,phi_fink_z6.phi.values)- np.power(10,phi_fink_z6.phi.values-phi
yerror_up=(np.power(10,phi_fink_z6.phi.values + phi_fink_z6.Bphi.values)- np.power(10
yerror_z6=np.array((yerror_low,yerror_up))
xerror_z6=(phi_fink_z6.BMag.values - phi_fink_z6.bMag.values)/2

yerror_low=np.power(10,phi_fink_z7.phi.values)- np.power(10,phi_fink_z7.phi.values-phi
yerror_up=(np.power(10,phi_fink_z7.phi.values + phi_fink_z7.Bphi.values)- np.power(10
yerror_z7=np.array((yerror_low,yerror_up))
xerror_z7=(phi_fink_z7.BMag.values - phi_fink_z7.bMag.values)/2

yerror_low=np.power(10,phi_fink_z8.phi.values)- np.power(10,phi_fink_z8.phi.values-phi
yerror_up=(np.power(10,phi_fink_z8.phi.values + phi_fink_z8.Bphi.values)- np.power(10
yerror_z8=np.array((yerror_low,yerror_up))
xerror_z8=(phi_fink_z8.BMag.values - phi_fink_z8.bMag.values)/2
```

In [11]:
```
fig, axs = plt.subplots(2, 3,sharex='col',sharey='row',gridspec_kw={'hspace': 0, 'wspa

(ax1,ax2,ax3),(ax4,ax5,ax6) = axs

#plt.figure(figsize=(50,10))
fig.suptitle("Función de Luminosidad UV", fontsize=24)

#Chart1
#ax1=plt.subplot(1,5,1)
ax1.plot(M,phi(M,4),"b-", label="Ajuste Scheter Finfelstein et al. (2015a)")
ax1.errorbar(phi_fink_z4.loc[:,["bMag","BMag"]].mean(axis=1).values,np.power(10,phi_fi
             yerr=yerror_z4, xerr=xerror_z4,marker='o',ms=10, mfc='b',ecolor="b",lines
ax1.plot(uvlf_z4["#UV_Center"],uvlf_z4["Number_Density"], "r-",label="Funcion de Lumin

ax1.set_title("$z=4$", fontsize=20)
ax1.set_ylabel("$\phi(Mag^{-1}Mpc^{-3})$", fontsize=16)
ax1.set_yscale("log")

#Chart2
#ax2=plt.subplot(1,5,2)
```

```
ax2.plot(M,phi(M,5),"b-", label="Ajuste Scheter Finfelstein et al. (2015a)")
ax2.errorbar(phi_fink_z5.loc[:,["bMag","BMag"]].mean(axis=1).values,np.power(10,phi_fi
             yerr=yerror_z5, xerr=xerror_z5,marker='o',ms=10, mfc='b',ecolor="b",lines
ax2.plot(uvlf_z5["#UV_Center"],uvlf_z5["Number_Density"], "r-",label="Funcion de Lumin

ax2.set_title("$z=5$", fontsize=20)
ax2.set_yscale("log")

#Chart3
#ax3=plt.subplot(1,5,3)
ax3.plot(M,phi(M,6),"b-", label="Ajuste Scheter Finfelstein et al. (2015a)")
ax3.errorbar(phi_fink_z6.loc[:,["bMag","BMag"]].mean(axis=1).values,np.power(10,phi_fi
             yerr=yerror_z6, xerr=xerror_z6,marker='o',ms=10, mfc='b',ecolor="b",lines
ax3.plot(uvlf_z6["#UV_Center"],uvlf_z6["Number_Density"], "r-",label="Funcion de Lumin

ax3.set_title("$z=6$", fontsize=20)
ax3.set_yscale("log")

#Chart4
#ax4=plt.subplot(1,5,4)
ax4.plot(M,phi(M,7),"b-", label="Ajuste Scheter Finfelstein et al. (2015a)")
ax4.errorbar(phi_fink_z7.loc[:,["bMag","BMag"]].mean(axis=1).values,np.power(10,phi_fi
             yerr=yerror_z7, xerr=xerror_z7,marker='o',ms=10, mfc='b',ecolor="b",lines
ax4.plot(uvlf_z7["#UV_Center"],uvlf_z7["Number_Density"], "r-",label="Funcion de Lumin
ax4.legend(fontsize=14)

ax4.set_title("$z=7$", fontsize=20)
ax4.set_xlabel("UV Mag", fontsize=16)
ax4.set_ylabel("$\phi(Mag^{-1}Mpc^{-3})$", fontsize=16)
ax4.set_yscale("log")

#Chart5
#ax5=plt.subplot(1,5,5)
ax5.plot(M,phi(M,8),"b-", label="Ajuste Scheter Finfelstein et al. (2015a)")
ax5.errorbar(phi_fink_z8.loc[:,["bMag","BMag"]].mean(axis=1).values,np.power(10,phi_fi
             yerr=yerror_z8, xerr=xerror_z8,marker='o',ms=10, mfc='b',ecolor="b",lines
ax5.plot(uvlf_z8["#UV_Center"],uvlf_z8["Number_Density"], "r-",label="Funcion de Lumin

ax5.set_title("$z=8$", fontsize=20)
ax5.set_xlabel("UV Mag", fontsize=16)
ax5.set_yscale("log")

ax6.set_xlabel("UV Mag", fontsize=16)

fig.savefig("charts/Func_UV.jpg")
fig.show()
```

```
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\matplotlib\figure
```

```
"matplotlib is currently using a non-GUI backend, "
```



$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$

```
In [ ]: gaus = lambda x,a,b,sigma:np.exp(-(x-b)**2/(2*sigma**2))*a

        def gaus_inv(y,a,b,sigma):
            a1=1
            b1=-2*b
            c1= b**2+2*sigma**2*np.log(y/a)
            return eq_2d(a1,b1,c1)

        def ajuste(x,y,mu):
            mean = np.sum(x*y)/np.sum(y)
            sigma = np.sqrt(np.sum(y*np.power(x-mean,2))/np.sum(y))
            return curve_fit(gaus,x,y,p0=[1,mu,sigma])

        def uvlf_hmf(hmf_z,uvlf_z, charts=False, ax=[]):
            df_itp=hmf_z.sort_values(["#Log10(HM)"], ascending=0)
            df_sample=uvlf_z.sort_values(["#UV_Center"], ascending=1)
```

```python
df_itp["Number_Density AC"]=df_itp["Number_Density"].cumsum()
df_sample["Number_Density AC"]=df_sample["Number_Density"].cumsum()

uv=df_sample["#UV_Center"].values
n_uv=df_sample["Number_Density"].values
hm=df_itp["#Log10(HM)"].values
n_hm=df_itp["Number_Density"].values

mask= hm>=10
hm=hm[mask]
n_hm=n_hm[mask]

mu_hm=hm[n_hm==n_hm.max()][0]
p_hm,_=ajuste(hm,n_hm,mu_hm)

mu_uv=uv[n_uv==n_uv.max()][0]
p_uv,_=ajuste(uv,n_uv,mu_uv)

s_uv=np.arange(uv.min(),uv.max()+1.1*(uv.max() - uv.min())/1000,(uv.max() - uv.min
s_hm=np.arange(hm.max(),hm.min()+1.1*(hm.min() - hm.max())/1000,(hm.min() - hm.max

def f_uv(x, inverse=False):
    f_z = lambda z:gaus(z,p_uv[0],p_uv[1],p_uv[2])
    # Integramos para obtener la densidad acumulada
    s_y = np.array([integrate.quad(f_z,-np.infty,i)[0] for i in s_uv])
    #s_y = f_z(s_uv)
     # Una vez que tenemos 1000 puntos interpolamos linealmente
    f_y= interp1d(s_y,s_uv)(x) if inverse else interp1d(s_uv,s_y)(x)
    return f_y


def f_hm (y,inverse=False):
    f_z = lambda z:gaus(z,p_hm[0],p_hm[1],p_hm[2])
    # Integramos para obtener la densidad acumulada
    s_y = np.array([integrate.quad(f_z,i,np.infty)[0] for i in s_hm])
    #s_y = f_z(s_hm)
    # Una vez que tenemos 1000 puntos interpolamos linealmente
    f_y= interp1d(s_y,s_hm)(y) if inverse else interp1d(s_hm,s_y)(y)
    return f_y

f = lambda x: f_hm(f_uv(x), True)
f_z_uv = lambda z:gaus(z,p_uv[0],p_uv[1],p_uv[2])
f_z_hm = lambda z:gaus(z,p_hm[0],p_hm[1],p_hm[2])

if charts:
    ax1,ax2,ax3,l = ax
    ax1.plot(uv,f_uv(uv), label=l)
    ax2.plot(hm,f_hm(hm),label=l)
```

```
            ax3.plot(s_uv,f(s_uv), label=l)
            return f,f_hm,f_uv,f_z_uv,f_z_hm
        else:
            return f,f_hm,f_uv,f_z_uv,f_z_hm
```

In [ ]:
```
fig = plt.figure(figsize=(15,7))
ax1=fig.add_subplot(221)
ax2=fig.add_subplot(223)
ax3=fig.add_subplot(122)
fig.text(0.04, 0.5, 'Densidad Numérica($>M_{Halo}/<M_{UV}$)[Mpc$^{-3}$]', va='center',
m_z=[m_z4,m_z5,m_z6,m_z7,m_z8]

print("------------------------------------------------------------")
print("Comienzo del Bucle.")
print("------------------------------------------------------------")

for i in range(5):
    l = "z=%i"%(i+4)
    print("------------------------------------------------------------")
    print(l)
    print("------------------------------------------------------------")

    f,f_hm,f_uv,f_z_uv,f_z_hm = uvlf_hmf(hmfs_list[i],uvlfs_list[i],True,[ax1,ax2,ax3,
    uv_lim = uvlfs_list[i]["#UV_Center"].max(),uvlfs_list[i]["#UV_Center"].min()
    m_halo = np.zeros(m_z[i].shape[0])
    mask = (m_z[i]["1500Mag"].values<=uv_lim[0]) & (m_z[i]["1500Mag"].values>=uv_lim[1
    mag_uv = m_z[i]["1500Mag"].values[mask]
    m_halo[mask] = f(mag_uv)
    m_halo[~mask] = np.nan
    m_z[i]["Mh"] = m_halo
    m_z[i]["N"] = f_hm(m_halo)
    m_z[i]["n_hm"] = f_z_hm(m_halo)
    m_z[i]["n_uv"] = f_z_uv(m_z[i]["1500Mag"].values)

print("------------------------------------------------------------")
print("Fin del Bucle")
print("------------------------------------------------------------")
print("------------------------------------------------------------")
print("Parametros de la Gráfica 1")
print("------------------------------------------------------------")

ax1.set_xlim(-19,-23)
ax1.set_ylim(10**(-7),10**(-2))
ax1.set_yscale("log")
ax1.xaxis.set_label_position('top')
ax1.xaxis.set_ticks_position("top")
ax1.set_xlabel("$M_{UV}$", fontsize=16)
ax1.xaxis.set_tick_params(labelsize=12)
```

```
        ax1.yaxis.set_tick_params(labelsize=12)

        print("------------------------------------------------------------")
        print("Parametros de la Gráfica 2")
        print("------------------------------------------------------------")
        ax2.set_xlim(10,13)
        ax2.set_ylim(10**(-7),10**(-2))
        ax2.set_yscale("log")
        ax2.set_xlabel("$\log(M_{Halo})$", fontsize=16)
        ax2.xaxis.set_tick_params(labelsize=12)
        ax2.yaxis.set_tick_params(labelsize=12)

        print("------------------------------------------------------------")
        print("Parametros de la Gráfica 3")
        print("------------------------------------------------------------")
        ax3.set_xlim(-22,-18)
        ax3.set_ylim(10,12.5)
        ax3.set_xlabel("$M_{UV}$", fontsize=16)
        ax3.set_ylabel("$\log(M_{Halo})$", fontsize=16)
        ax3.xaxis.set_tick_params(labelsize=12)
        ax3.yaxis.set_tick_params(labelsize=12)
        #ax3.yaxis.set_label_position('right')
        ax3.yaxis.set_ticks_position("right")
        ax3.legend(fontsize=14)

        plt.savefig("charts/Fink_halo_uv.jpg")
        print("------------------------------------------------------------")
        print("Gráfica Guardada.")
        print("------------------------------------------------------------")
```

```
------------------------------------------------------------
Comienzo del Bucle.
------------------------------------------------------------
------------------------------------------------------------
z=4
------------------------------------------------------------


C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/i
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\scipy\interpolate
  below_bounds = x_new < self.x[0]
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\scipy\interpolate
  above_bounds = x_new > self.x[-1]
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/i
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/i
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/i


------------------------------------------------------------
z=5
------------------------------------------------------------
```

```
In [ ]: # Z=7
        f= uvlf_hmf(hmfs_list[3],uvlfs_list[3])[0]
        f(-21)
```

Comparar con el HMF

```
In [15]: !head -n 12 HMFCalc/Seth_WMAP7/mVector_z\=4\ .txt
```

```
"head" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

```
In [16]: #Parametros WMAP7
         h = 0.704
```

```
In [17]: archivos = ["HMFCalc/Seth_WMAP7/mVector_z=4 .txt","HMFCalc/Seth_WMAP7/mVector_z=5 .tx
                     "HMFCalc/Seth_WMAP7/mVector_z=6 .txt","HMFCalc/Seth_WMAP7/mVector_z=7 .txt
                     "HMFCalc/Seth_WMAP7/mVector_z=8 .txt"]
         hmfcalc_list=[]

         for a in archivos:
             hmfcalc_z=pd.read_csv(a,sep="\s",header=None, skiprows=12)
             hmfcalc_z = hmfcalc_z.iloc[:,[0,7,11]]
             hmfcalc_z=hmfcalc_z.rename(columns={0:"M_h",7:"dn/dlnm",11:"Lbox"})
             hmfcalc_z["M_h"]=np.log10(hmfcalc_z["M_h"].values*h)
             hmfcalc_list.append(hmfcalc_z)
```

```
C:\Users\santiago.arranz\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launche
  import sys
```

```
---------------------------------------------------------------------------

FileNotFoundError                         Traceback (most recent call last)

<ipython-input-17-2b11b1c30e85> in <module>()
      5
      6 for a in archivos:
----> 7     hmfcalc_z=pd.read_csv(a,sep="\s",header=None, skiprows=12)
      8     hmfcalc_z = hmfcalc_z.iloc[:,[0,7,11]]
      9     hmfcalc_z=hmfcalc_z.rename(columns={0:"M_h",7:"dn/dlnm",11:"Lbox"})


~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\io\parsers.py in parser_f
    683         )
    684
--> 685         return _read(filepath_or_buffer, kwds)
    686
    687     parser_f.__name__ = name


~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\io\parsers.py in _read(fil
    455
    456     # Create the parser.
--> 457     parser = TextFileReader(fp_or_buf, **kwds)
    458
    459     if chunksize or iterator:


~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__
    893             self.options["has_index_names"] = kwds["has_index_names"]
    894
--> 895         self._make_engine(self.engine)
    896
    897     def close(self):


~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\io\parsers.py in _make_eng
   1145                     ' "python-fwf")'.format(engine=engine)
   1146                 )
-> 1147             self._engine = klass(self.f, **self.options)
   1148
   1149     def _failover_to_python(self):


~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__
   2291             encoding=self.encoding,
   2292             compression=self.compression,
```

```
-> 2293             memory_map=self.memory_map,
   2294         )
   2295         self.handles.extend(handles)


    ~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\io\common.py in _get_handl
    400         elif is_text:
    401             # No explicit encoding
--> 402             f = open(path_or_buf, mode, errors="replace", newline="")
    403         else:
    404             # Binary mode


    FileNotFoundError: [Errno 2] No such file or directory: 'HMFCalc/Seth_WMAP7/mVector_z=4
```

In [ ]:
```
plt.figure(figsize=(10,10))
z=4
dic={}
col_dic=[]
colors=["b","y","g","r","k"]
for m in m_z:
    hmfcalc_z=hmfcalc_list[z-4]
    m_array=m["Mh"].values
    n_array=m["N"].values
    n_array = n_array[~np.isnan(m_array)]
    m_array = m_array[~np.isnan(m_array)]
    l="z=%i"%(z)
    m_sample = np.arange(m_array.min(),m_array.max()+0.05,0.05)
    cuentas = np.array([ np.sum(n_array[(m_array >= m_i-0.025) & (m_array<m_i+0.025)])
    n_array=cuentas/np.sum(cuentas)
    plt.plot(hmfcalc_z.M_h,hmfcalc_z["dn/dlnm"], label="HMF z=%i"%(z), color=colors[z-4
    plt.scatter(m_sample,cuentas,c=colors[z-4],label=l)
    z+=1

plt.yscale("log")
plt.xlim(10,13)
plt.ylim(10**(-10),10**(-0))
plt.legend()
```

In [ ]:
```
f,f_hm,f_uv,f_z_uv,f_z_hm= uvlf_hmf(hmfs_list[i],uvlfs_list[i])
uv_lim = uvlfs_list[i]["#UV_Center"].max(),uvlfs_list[i]["#UV_Center"].min()
mask = (m_z[i]["1500Mag"].values<=uv_lim[0]) & (m_z[i]["1500Mag"].values>=uv_lim[1])
mag_uv = m_z[i]["1500Mag"].values[mask]
mag_uv.sort()
plt.plot(mag_uv,f_z_uv(mag_uv),label="Gauss")
plt.plot(mag_uv,f_uv(mag_uv),label="Integral")
plt.legend()
```

```
In [ ]: f,f_hm,f_uv,f_z_uv,f_z_hm= uvlf_hmf(hmfs_list[i],uvlfs_list[i])
        m_halo = m_z7['Mh'].values
        m_halo = m_halo[~np.isnan(m_halo)]
        m_halo.sort()
        plt.plot(m_halo,f_z_hm(m_halo))
        plt.plot(m_halo,f_hm(m_halo))
        plt.plot(m_halo,f_z_hm(m_halo)[-1::-1].cumsum()[-1::-1]/(m_halo[-1]-m_halo[0]))
```

```
In [38]: cuentas/cuentas.sum()
```

```
Out[38]: array([0.07914573, 0.69472362, 0.22361809, 0.00251256])
```

1.3 Distribución de los datos de Section 2.1

```
In [185]: !head -n 29 "Behroozi/sfr_catalog_0.073623.txt"
```

```
#ID DescID UPID Flags Uparent_Dist X Y Z VX VY VZ M V MP VMP R Rank1 Rank2 RA RARank SM ICL SFR

#Columns:

#ID: Unique halo ID

#DescID: ID of descendant halo (or -1 at z=0).

#UPID: -1 for central halos, otherwise, ID of largest parent halo

#Flags: Ignore

#Uparent_Dist: Ignore

#X Y Z: halo position (comoving Mpc/h)

#VX VY VZ: halo velocity (physical peculiar km/s)

#M: Halo mass (Bryan & Norman 1998 virial mass, Msun)

#V: Halo vmax (physical km/s)

#MP: Halo peak historical mass (BN98 vir, Msun)

#VMP: Halo vmax at the time when peak mass was reached.

#R: Halo radius (BN98 vir, comoving kpc/h)

#Rank1: halo rank in Delta_vmax (see UniverseMachine paper)

#Rank2, RA, RARank: Ignore

#SM: True stellar mass (Msun)
```

```
#ICL: True intracluster stellar mass (Msun)

#SFR: True star formation rate (Msun/yr)

#Obs_SM: observed stellar mass, including random & systematic errors (Msun)

#Obs_SFR: observed SFR, including random & systematic errors (Msun/yr)

#SSFR: observed SSFR

#SMHM: SM/HM ratio

#Obs_UV: Observed UV Magnitude (M_1500 AB)

#h = 0.680000

#Omega_Matter = 0.307115

#Omega_Lambda = 0.692885

#Box size: 250.000000 Mpc/h

#Note: all halo masses converted to be in Msun, not Msun/h.
```

In [184]:
```python
bolshoi = pd.read_csv("Behroozi/sfr_catalog_0.073623.txt", sep="\s", skiprows=29)
bolshoi
```

```
/Users/santi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: ParserWarning: Fal
  """Entry point for launching an IPython kernel.
```

Out[184]:

	#ID	DescID	UPID	Flags	Uparent_Dist	X	Y	\
0	405434	627025	-1	2	2.0	10.700050	24.257339	
1	405437	627028	-1	2	2.0	9.471750	24.515860	
2	405439	627033	-1	2	2.0	9.332960	24.412491	
3	405629	627319	-1	3	2.0	18.271070	18.654341	
4	405630	627320	-1	3	2.0	18.023230	19.298660	
5	405631	627323	-1	3	2.0	17.358450	18.844521	
6	405634	627326	-1	2	2.0	17.496620	19.623699	
7	405637	627329	-1	2	2.0	17.569040	19.675470	
8	405639	627332	-1	3	2.0	19.749479	17.685520	
9	405640	627333	-1	2	2.0	16.221670	19.606350	
10	405643	627340	-1	3	2.0	17.925150	19.677549	
11	405644	627341	-1	2	2.0	15.759720	19.406200	
12	405647	627345	-1	3	2.0	16.236280	19.292910	

13	405651	627349	−1	2	2.0	17.063919	20.770081
14	405652	627350	−1	2	2.0	17.111231	20.669121
15	405656	627354	−1	2	2.0	16.744200	20.233841
16	405658	627359	−1	3	2.0	16.532221	18.315069
17	405659	627361	−1	2	2.0	15.720520	18.113590
18	405450	627056	−1	2	2.0	19.467720	22.720150
19	405451	627057	−1	3	2.0	19.283541	22.747841
20	405455	627064	−1	2	2.0	19.843451	23.026739
21	405456	627066	−1	3	2.0	19.793690	24.796419
22	405458	627070	−1	2	2.0	21.145189	24.568859
23	405465	627079	−1	2	2.0	20.346390	20.786640
24	405468	627082	−1	2	2.0	23.121710	24.201170
25	405477	627092	−1	2	2.0	20.409590	24.231850
26	405372	626932	−1	3	2.0	8.063740	13.751390
27	405373	626933	−1	3	2.0	8.193750	12.635750
28	405375	626937	−1	3	2.0	10.595200	9.496620
29	405311	626833	−1	3	2.0	6.253390	17.666281
...	...	...	...	...	...	...	...
49729	552338	841138	−1	2	2.0	236.675781	245.663208
49730	552080	840779	−1	2	2.0	242.971603	230.308411
49731	551662	840196	−1	2	2.0	234.428070	245.642624
49732	552081	840783	−1	3	2.0	236.942642	230.703583
49733	552470	841316	−1	2	2.0	237.501617	241.251740
49734	552403	841226	−1	3	2.0	245.157501	225.059845
49735	551702	840255	−1	3	2.0	239.041992	229.057938
49736	552399	841220	−1	2	2.0	236.794861	220.765640
49737	552472	841318	−1	2	2.0	236.850647	228.791946
49738	552475	841323	−1	2	2.0	237.629089	229.768188
49739	552476	841324	−1	3	2.0	237.555389	229.677231
49740	552477	841326	−1	3	2.0	232.741180	223.861374
49741	552346	841150	−1	3	2.0	233.267288	245.664520
49742	552473	841321	−1	2	2.0	230.049118	232.544464
49743	552085	840791	−1	3	2.0	226.741501	222.580048
49744	552347	841151	−1	2	2.0	237.949265	239.613220
49745	552087	840793	−1	2	2.0	226.421143	237.418213
49746	552474	841322	−1	2	2.0	231.129364	244.472443
49747	551909	840531	−1	2	2.0	239.567703	237.952179
49748	551908	840530	−1	2	2.0	238.171402	241.327011
49749	551636	840163	−1	2	2.0	248.076782	249.612381
49750	551637	840164	−1	2	2.0	248.136856	249.773911
49751	552333	841131	−1	3	2.0	233.877045	239.784256
49752	551910	840532	−1	3	2.0	228.386856	235.010178
49753	551594	840103	−1	3	2.0	248.087616	247.773087
49754	551715	840272	−1	2	2.0	226.923767	227.495514
49755	552453	841295	−1	2	2.0	228.060257	230.137009
49756	551618	840136	−1	2	2.0	238.360291	247.365601
49757	551664	840202	−1	3	2.0	233.974640	227.820953
49758	551693	840242	−1	2	2.0	232.234436	247.629517

	Z	VX	VY	...	RA	RARank	\
0	29.222601	-8.670000	45.330002	...	1.213039	1.541433	
1	28.488449	63.500000	15.510000	...	-0.873632	0.101669	
2	26.648420	41.970001	-21.020000	...	-0.649631	0.570677	
3	46.963711	-120.660004	161.929993	...	-1.444936	-1.444936	
4	47.493092	-109.010002	73.639999	...	-0.447806	-0.447806	
5	46.709560	-15.330000	159.169998	...	1.906949	1.906949	
6	47.060810	-47.169998	63.680000	...	-0.939994	-1.048224	
7	49.013000	-46.430000	67.080002	...	-1.260807	-0.919974	
8	46.604599	-138.339996	160.080002	...	1.026307	1.026307	
9	48.948502	47.189999	70.290001	...	0.957268	1.164022	
10	46.351570	-136.179993	89.129997	...	1.057757	1.057757	
11	44.983829	36.619999	165.860001	...	-1.409150	-0.344358	
12	45.098091	-18.650000	161.649994	...	0.023849	0.023849	
13	46.173222	-119.120003	-27.510000	...	-0.932100	-0.533534	
14	46.179192	-115.150002	-10.070000	...	1.760944	1.619551	
15	45.733021	-55.720001	73.110001	...	1.140937	1.637054	
16	48.891270	22.170000	155.080002	...	0.182622	0.182622	
17	44.782490	60.750000	220.240005	...	-1.562948	-2.763681	
18	23.842051	-135.100006	35.630001	...	1.877243	1.474318	
19	23.729130	-80.800003	27.629999	...	-1.191057	-1.191057	
20	22.840389	-100.150002	47.959999	...	-0.134840	0.122117	
21	26.645691	-54.779999	-66.820000	...	-0.338116	-0.338116	
22	25.462299	-143.000000	32.320000	...	0.011877	0.107585	
23	27.937389	-69.370003	157.910004	...	0.190902	-0.145710	
24	29.176741	-151.789993	9.480000	...	0.764226	1.314273	
25	31.573271	-19.690001	29.770000	...	-1.096920	-1.389619	
26	12.362870	78.260002	22.020000	...	0.160758	0.160758	
27	11.968750	72.169998	68.040001	...	-1.851160	-1.851160	
28	11.785330	22.540001	64.669998	...	-0.764701	-0.764701	
29	18.936220	77.940002	179.940002	...	-0.468288	-0.468288	
...	...	...	...	...	...	...	
49729	234.666489	-30.180000	-20.139999	...	-1.808135	-1.830193	
49730	229.020889	-25.260000	-88.589996	...	-0.057139	-0.711408	
49731	212.007507	74.389999	-13.630000	...	-0.132456	0.473144	
49732	229.279922	29.030001	-76.550003	...	-0.849766	-0.849766	
49733	242.113953	71.830002	-78.629997	...	-0.963537	-0.250612	
49734	0.034750	23.629999	92.930000	...	1.397298	1.397298	
49735	193.493423	128.740005	-20.320000	...	0.835161	0.835161	
49736	1.052130	72.779999	157.130005	...	-1.021031	-1.539269	
49737	0.772500	154.190002	-9.380000	...	0.050421	0.177023	
49738	1.627990	106.949997	-5.320000	...	0.179255	-0.548809	
49739	1.636200	126.680000	-5.320000	...	-0.403217	-0.403217	
49740	242.379150	114.550003	5.200000	...	-1.660235	-1.660235	
49741	227.723282	26.440001	-2.040000	...	1.783372	1.783372	
49742	245.177063	81.209999	-31.219999	...	-1.091418	-1.394530	
49743	224.917206	211.149994	88.660004	...	0.382201	0.382201	

```
49744  227.766312  -73.610001 -131.229996  ...  0.216940  0.228817
49745  226.898209  126.580002  -31.040001  ... -0.761650 -0.481956
49746  248.374557  -14.590000  -58.619999  ... -1.132617 -1.614506
49747  214.232605  -13.190000  -74.849998  ...  1.129535  0.844901
49748  212.446014  -42.720001   26.610001  ...  0.254348 -0.042153
49749  212.451584  -96.440002  -67.860001  ... -0.514334 -0.286179
49750  212.458359  -92.699997  -80.180000  ... -1.770034 -1.331889
49751  232.279007   66.760002  -10.340000  ...  1.217798  1.217798
49752  208.970917   63.270000    4.340000  ... -0.606096 -0.606096
49753  205.136169  -28.840000  -57.830002  ... -0.121774 -0.121774
49754  203.444214  111.250000   49.290001  ...  1.179133  0.932860
49755  245.545898  105.459999   -7.090000  ...  0.298778  0.584280
49756  210.265533   42.779999  -56.669998  ... -0.139261  1.287020
49757  200.062637  154.210007  -12.800000  ... -0.647312 -0.647312
49758  207.666626   50.110001   31.209999  ... -0.092928 -0.074545

                SM  ICL       SFR       obs_SM   obs_SFR          SSFR  \
0       10190000.0  0.0  0.678400   11370000.0  0.645200  5.676000e-08
1        4483000.0  0.0  0.347000     461600.0  0.330000  7.149000e-07
2       17150000.0  0.0  0.187600   16360000.0  0.178400  1.091000e-08
3         855400.0  0.0  0.115200    1105000.0  0.109500  9.916000e-08
4          52220.0  0.0  0.007031      50920.0  0.006687  1.313000e-07
5          58700.0  0.0  0.007904      50370.0  0.007517  1.492000e-07
6         241900.0  0.0  0.001367     345600.0  0.001300  3.763000e-09
7        2711000.0  0.0  0.097960    1496000.0  0.093170  6.226000e-08
8          99420.0  0.0  0.013390     191900.0  0.012730  6.633000e-08
9      108300000.0  0.0  2.200000  117300000.0  2.092000  1.783000e-08
10        226000.0  0.0  0.030420      46580.0  0.028940  6.212000e-07
11      12560000.0  0.0  0.603800    7717000.0  0.574300  7.442000e-08
12          7712.0  0.0  0.001038       4456.0  0.000988  2.216000e-07
13       1566000.0  0.0  0.044160     718700.0  0.042000  5.843000e-08
14       5284000.0  0.0  0.314300    1802000.0  0.298900  1.659000e-07
15        904600.0  0.0  0.056150     649500.0  0.053400  8.223000e-08
16        344000.0  0.0  0.046320     521000.0  0.044050  8.456000e-08
17        125200.0  0.0  0.005629      66620.0  0.005354  8.037000e-08
18       5382000.0  0.0  0.518700    6945000.0  0.493300  7.103000e-08
19        158100.0  0.0  0.021280     196400.0  0.020240  1.030000e-07
20       2941000.0  0.0  0.135400    4883000.0  0.128800  2.637000e-08
21          9309.0  0.0  0.001253      10390.0  0.001192  1.147000e-07
22       7169000.0  0.0  0.329300    6494000.0  0.313100  4.822000e-08
23        338700.0  0.0  0.020650     442100.0  0.019640  4.444000e-08
24       4006000.0  0.0  0.286700    2896000.0  0.272600  9.414000e-08
25       8861000.0  0.0  0.545000    4223000.0  0.518300  1.227000e-07
26        137000.0  0.0  0.018450      88260.0  0.017550  1.988000e-07
27         47630.0  0.0  0.006413      29330.0  0.006099  2.079000e-07
28         65090.0  0.0  0.008764      28950.0  0.008335  2.879000e-07
29         16300.0  0.0  0.002194      14700.0  0.002087  1.420000e-07
...            ...  ...       ...          ...       ...           ...
```

49729	1861000.0	0.0	0.087750	961000.0	0.083460	8.684000e-08
49730	2872000.0	0.0	0.202900	1467000.0	0.192900	1.315000e-07
49731	3004000.0	0.0	0.127200	1028000.0	0.121000	1.177000e-07
49732	140100.0	0.0	0.018860	72400.0	0.017940	2.478000e-07
49733	1166000.0	0.0	0.064480	705600.0	0.061320	8.691000e-08
49734	51850.0	0.0	0.006981	49800.0	0.006639	1.333000e-07
49735	2602000.0	0.0	0.350400	998600.0	0.333200	3.337000e-07
49736	1880000.0	0.0	0.072130	2829000.0	0.068600	2.425000e-08
49737	2143000.0	0.0	0.110800	920600.0	0.105400	1.145000e-07
49738	291700.0	0.0	0.025060	211900.0	0.023840	1.125000e-07
49739	13180.0	0.0	0.001775	21240.0	0.001688	7.949000e-08
49740	86440.0	0.0	0.011640	36530.0	0.011070	3.030000e-07
49741	544800.0	0.0	0.073360	368500.0	0.069770	1.893000e-07
49742	782500.0	0.0	0.033220	436200.0	0.031600	7.244000e-08
49743	257600.0	0.0	0.034680	440000.0	0.032980	7.496000e-08
49744	986100.0	0.0	0.031650	336500.0	0.030100	8.946000e-08
49745	1339000.0	0.0	0.126300	378900.0	0.120200	3.171000e-07
49746	769200.0	0.0	0.035250	587100.0	0.033520	5.710000e-08
49747	1866000.0	0.0	0.151200	2078000.0	0.143800	6.922000e-08
49748	207800.0	0.0	0.017190	123100.0	0.016350	1.328000e-07
49749	81910.0	0.0	0.006204	49420.0	0.005901	1.194000e-07
49750	667800.0	0.0	0.010340	741500.0	0.009837	1.327000e-08
49751	2904000.0	0.0	0.391000	2694000.0	0.371800	1.380000e-07
49752	43580.0	0.0	0.005867	82750.0	0.005580	6.743000e-08
49753	69760.0	0.0	0.009393	27210.0	0.008933	3.283000e-07
49754	2567000.0	0.0	0.238000	365300.0	0.226300	6.197000e-07
49755	232700.0	0.0	0.014080	132200.0	0.013390	1.013000e-07
49756	549800.0	0.0	0.024790	149600.0	0.023580	1.577000e-07
49757	68920.0	0.0	0.009279	30360.0	0.008825	2.907000e-07
49758	47580000.0	0.0	0.553700	103700000.0	0.526600	5.077000e-09

	SM/HM	obs_UV
0	0.002982	-17.749
1	0.000678	-17.026
2	0.002688	-17.188
3	0.000289	-15.700
4	0.000046	-12.717
5	0.000129	-12.843
6	0.000265	-13.109
7	0.000661	-16.122
8	0.000145	-13.410
9	0.011320	-19.129
10	0.000165	-14.290
11	0.002506	-17.704
12	0.000017	-10.648
13	0.000572	-15.332
14	0.001449	-16.947
15	0.000794	-15.231

```
16     0.000137 -14.738
17     0.000061 -12.956
18     0.001161 -17.344
19     0.000063 -13.908
20     0.000614 -16.316
21     0.000020 -10.852
22     0.001851 -17.107
23     0.000212 -14.188
24     0.001098 -16.850
25     0.001341 -17.530
26     0.000086 -13.755
27     0.000030 -12.618
28     0.000048 -12.954
29     0.000024 -11.459
...         ...      ...
49729  0.000340 -15.818
49730  0.000788 -16.487
49731  0.001014 -16.110
49732  0.000154 -13.779
49733  0.000394 -15.445
49734  0.000114 -12.709
49735  0.000672 -16.846
49736  0.000634 -15.640
49737  0.000672 -15.977
49738  0.000151 -14.241
49739  0.000029 -11.229
49740  0.000034 -13.260
49741  0.000398 -15.225
49742  0.000229 -14.890
49743  0.000161 -14.430
49744  0.000618 -14.859
49745  0.000452 -15.910
49746  0.000211 -14.909
49747  0.000744 -16.160
49748  0.000152 -13.853
49749  0.000103 -12.795
49750  0.000293 -14.366
49751  0.000708 -16.957
49752  0.000048 -12.522
49753  0.000061 -13.029
49754  0.000704 -16.551
49755  0.000255 -13.687
49756  0.000483 -14.544
49757  0.000060 -13.016
49758  0.008700 -18.280

[49759 rows x 28 columns]
```

1.4 Función de Masa de Halo Teórica

Las predicciones teóricas de la función de masa de halo presentadas en Section 2.1 fueron calculadas a través de HMFCalc (Section 2.1) usando las estimaciones de Section 2.1.

Para comenzar, (Section 2.1.) he instalado tanto la libreria hmf como la app-web diseñada con el framework Django HMFcalc (1). Ésta última me ha parecido una muy buena opción debido a que su uso es bastante sencilla y que la integración de los datos de Section 2.1 creo que se podría realizar con mayor facilidad, sin embargo presenta una versión mucho menos actualizada de *hmf* el cual es el código base de la parte de cálculos de la app-web. De todas formas, se estudiarán la integración de los nuevos datos en ambas opciones con la intención de poder compartir las versiones en sus respectivos repositorios de GitHub, aunque los cálculos finales se realizarán con la última versión de *hmf*.(2)

---

(1) La versión de Django usada en esta aplicación es la 1.4, donde el código de la app web es solo compatible para las versiones de Python 2.x.

(2) Otra de las opciones para el cálculo de la función de masa de halo (Section 2.1) la presenta la librería de python yt utilizada en la explotación de los datos de RAMSES.

2 Notas

Parece que los datos de Behroozi no aportan nuevas parámetros de entrada para el HMF de Murray. Esto tirá un poco al traste lo pensado hasta ahora. Sin embargo, el HMF de Steindarht usa la función de ajuste de Seth, la cual parece estar basada en un universo Eisntein-deSitter y eso me ha llevado a pensar si se están usando las suposiciones de los parámetros correctas. Podemos intentar usar Reed2003 u otras funciones añadidas a posteriori que puedan encajar mejor £Cómo cambiaría los datos.?

Por otro lado me he descargado los datos de Behroozi y CANDELS

```
In [10]: !head ../../CANDELS/test.dat
```

```
#RA Dec Z(los) Z(cosmo) Scale ID DescID UPID Flags Uparent_Dist X Y Z VX VY VZ M V MP VMP R Ra

150.175095 2.018628 9.880285 9.877687 0.092748 6851068 9152134 -1 3 2.000000 6534.355469 -5.020

150.172134 2.022606 9.878959 9.878014 0.092748 6851070 9152137 -1 3 2.000000 6534.407715 -4.683

150.207108 2.023004 9.863716 9.860533 0.092748 6851107 9152181 -1 2 2.000000 6531.767090 -8.660

150.207870 2.024081 9.863198 9.860908 0.092748 6851109 9152183 -1 2 2.000000 6531.824219 -8.753

150.209305 2.022477 9.866434 9.864197 0.092748 6851110 9152184 -1 2 2.000000 6532.319336 -8.916

150.216507 2.023422 9.882704 9.881249 0.092748 6851111 9152185 -1 3 2.000000 6534.889160 -9.741

150.215530 2.034154 9.851378 9.851107 0.092748 6851679 9152903 -1 2 2.000000 6530.348145 -9.625
```

```
150.142242 2.007869 9.877820 9.876465 0.092748 7042004 9393767 -1 2 2.000000 6534.166992 -1.273

150.228592 2.038660 9.908158 9.907624 0.092748 6851477 9152644 -1 2 2.000000 6538.860840 -11.12
```

2.1 Referencias

- Steinhardt, C. L., Capak, P., Masters, D., & Speagle, J. S. (2016). The impossibly early galaxy problem. *The Astrophysical Journal, 824(1)*, 21.

- Steinhardt, C. L., & Speagle, J. S. (2014). A uniform history for galaxy evolution. *The Astrophysical Journal, 796(1)*, 25.

- Behroozi, P., Wechsler, R. H., Hearin, A. P., & Conroy, C. (2019). UniverseMachine: The correlation between galaxy growth and dark matter halo assembly from z= 0 10. *Monthly Notices of the Royal Astronomical Society, 488(3)*, 3143-3194.

- Murray, S. G., Power, C., & Robotham, A. S. G. (2013). HMFcalc: An online tool for calculating dark matter halo mass functions. *Astronomy and Computing, 3*, 23-34.

- Sheth, R. K., Mo, H. J., & Tormen, G. (2001). Ellipsoidal collapse and an improved model for the number and spatial distribution of dark matter haloes. *Monthly Notices of the Royal Astronomical Society, 323(1)*, 1-12.

- Hildebrandt, H., Pielorz, J., Erben, T., van Waerbeke, L., Simon, P., & Capak, P. (2009). CARS: the CFHTLS-Archive-Research Survey-II. Weighing dark matter halos of Lyman-break galaxies at z= 3–5. *Astronomy & Astrophysics, 498(3)*, 725-736.

- a) Bouwens, R. J., Illingworth, G. D., Oesch, P. A., Caruana, J., Holwerda, B., Smit, R., & Wilkins, S. (2015). Reionization after Planck: the derived growth of the cosmic ionizing emissivity now matches the growth of the galaxy UV luminosity density. *The Astrophysical Journal, 811(2)*, 140.

- b) Bouwens, R. J., Illingworth, G. D., Oesch, P. A., Trenti, M., Labbé, I., Bradley, L., ... & Franx, M. (2015). UV luminosity functions at redshifts z 4 to z 10: 10,000 galaxies from HST legacy fields. *The Astrophysical Journal, 803(1)*, 34.

- a) Finkelstein, S. L., Ryan Jr, R. E., Papovich, C., Dickinson, M., Song, M., Somerville, R. S., ... & Ashby, M. L. (2015). The evolution of the galaxy rest-frame ultraviolet luminosity function over the first two billion years. The Astrophysical Journal, 810(1), 71.

- b) Finkelstein, S. L., Song, M., Behroozi, P., Somerville, R. S., Papovich, C., Milosavljevi, M., ... & Fazio, G. G. (2015). An increasing stellar baryon fraction in bright galaxies at high redshift. The Astrophysical Journal, 814(2), 95.

- Wang, T., Schreiber, C., Elbaz, D., Yoshimura, Y., Kohno, K., Shu, X., ... & Lim, C. F. (2019). A dominant population of optically invisible massive galaxies in the early Universe. Nature, 572(7768), 211-214.