Ejercicio 1 : 1 Set de datos / Histograma

Antes de comenzar a correr cosas en este notebook, comentar que de principio a fin, en Mathematica Online, se necesitaron 6 minutos y medio en completar la evaluación completa.

In[19]:=
```
beginning = Now
            ahora
SetDirectory[NotebookDirectory[]]
establece direct··· directorio de cuaderno
```

Out[19]= Thu 14 Nov 2024 07:01:01 GMT−3

Out[20]= C:\Users\ELXMA\Desktop\REMONTADA\mamas\13.11

Datos de cuásares, ajuste a magnitudes en varias bandas.

In[21]:=
```
rawdata = Import["astrostatistics.psu.edu_MSMA_datasets_SDSS_QSO.dat"];
          importa
```

··· Import: File astrostatistics.psu.edu_MSMA_datasets_SDSS_QSO.dat not found during Import.

```
header = rawdata[[1]]
```

Out[•]= {SDSS, z, u_mag, sig_u_mag, g_mag, sig_g_mag, r_mag,
sig_r_mag, i_mag, sig_i_mag, z_mag, sig_z_mag, FIRST, ROSAT, Mp}

Buscar datos de magnitudes en r_, g_ y u_.

```
Position[header, "r_mag"]
posición
Position[header, "g_mag"]
posición
Position[header, "u_mag"]
posición
```

Out[•]= {{7}}

Out[•]= {{5}}

Out[•]= {{3}}

Seleccionar datos en los que todas las bandas tengan valores mayores a 15:

```
data = Select[rawdata,
       selecciona
  NumberQ[#[[3]]] && NumberQ[#[[5]]] && NumberQ[#[[7]]] && #[[3]] > 15 && #[[5]] > 15 && #[[7]] > 15 &]
  ¿número?          ¿número?          ¿número?
```

Out[•]=
```
{{000006.53+003055.2, 1.8227, 20.389, 0.066, 20.468,
  0.034, 20.332, 0.037, 20.099, 0.041, 20.053, 0.121, 0., -9., -25.1},
 {000008.13+001634.6, 1.8365, 20.233, 0.054, 20.2, 0.024, 19.945, 0.032,
  19.491, 0.032, 19.191, 0.068, 0., -9., -25.738}, ··· 77 289 ···,
 {235959.44+000841.5, 1.3553, 21.012, 0.113, 20.892, 0.038, 20.413,
  0.035, 20.451, 0.048, 20.103, 0.15, 0., -9., -24.064}}
```

$Failed ($Failed34 MB)

Primero, trabajar con datos de banda u, encontrar una distribución con FindDistribution:

```
ufilter = data[[All, 3]];
          todo
udist = SmoothKernelDistribution[ufilter];
        distribución de núcleo suave
bestdistu = FindDistribution[ufilter]
            encuentra distribución
```

Out[•]=
```
MixtureDistribution[{0.920482, 0.0795176},
 {LogisticDistribution[19.5795, 0.450639], LogNormalDistribution[3.14774, 0.0537833]}]
```

Ajustar una distribución bimodal, comparar numéricamente y gráficamente.

In[•]:=
```
umodel =
  MixtureDistribution[{a, b}, {NormalDistribution[u1, s1], NormalDistribution[u2, s2]}];
  distribución mezcla              distribución normal              distribución normal
```

In[•]:=
```
Parallelize[mincuad =
paraleliza
    Sum[(PDF[udist, ufilter[[k]]] - PDF[umodel, ufilter[[k]]])^2, {k, Length@ufilter}]];
    suma  función de densidad de probabil··  función de densidad de probabilidad  longitud
```

••• Parallelize: No parallel kernels available; proceeding with sequential evaluation.

```
solOLSu = AbsoluteTiming[
          duración absoluta
  NMinimize[mincuad, {{a, 0., 1.}, {b, 0., 1.}, {u1, 18., 20.},
  minimiza aproximadamente
    {s1, 0., 2.}, {u2, 19., 26.}, {s2, 0., 1.8}}, Method → "SimulatedAnnealing"]]
                                                   método
```

Out[•]=
```
{36.7161, {23.7274,
  {a → 0.160499, b → 1.08508, u1 → 19.3432, s1 → 0.257516, u2 → 19.6519, s2 → 0.991188}}}
```

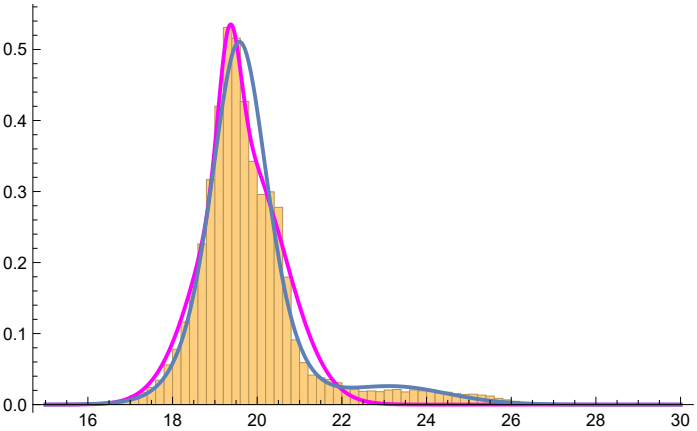
In[ ]:=
```
mincuadbest = Sum[(PDF[udist, ufilter[[k]]] - PDF[bestdistu, ufilter[[k]]])^2,
                  suma  función de densidad de probabil··  función de densidad de probabilidad
    {k, Length@ufilter}] // Parallelize
                          longitud                          paraleliza
```

Parallelize: No parallel kernels available; proceeding with sequential evaluation.

Out[ ]=
```
179.964
```

```
Show[
muestra
Histogram[ufilter, {15, 30, 0.2}, "PDF", PlotRange → All],
histograma                          func···  rango de rep···  todo
Plot[PDF[umodel /. solOLSu[[2, 2]], x], {x, 15, 30}, PlotStyle → Magenta, PlotRange → All],
repr···  función de densidad de probabilidad                  estilo de repr···  magenta  rango de rep···  todo
Plot[PDF[bestdistu, x], {x, 15, 30}, PlotRange → All]
repr···  función de densidad de probabilidad          rango de rep···  todo
]
```

Out[ ]=



Ajustar una distribución tri-modal, comparar con las dos anteriores.

In[ ]:=
```
umodel2 = MixtureDistribution[{a, b, c},
                              distribución mezcla
    {NormalDistribution[u1, s1], NormalDistribution[u2, s2], NormalDistribution[u3, s3]}];
     distribución normal              distribución normal              distribución normal
```

In[ ]:=
```
Parallelize[mincuad2 =
paraleliza
    Sum[(PDF[udist, ufilter[[k]]] - PDF[umodel2, ufilter[[k]]])^2, {k, Length@ufilter}]];
    suma  función de densidad de probabil··  función de densidad de probabilidad          longitud
```

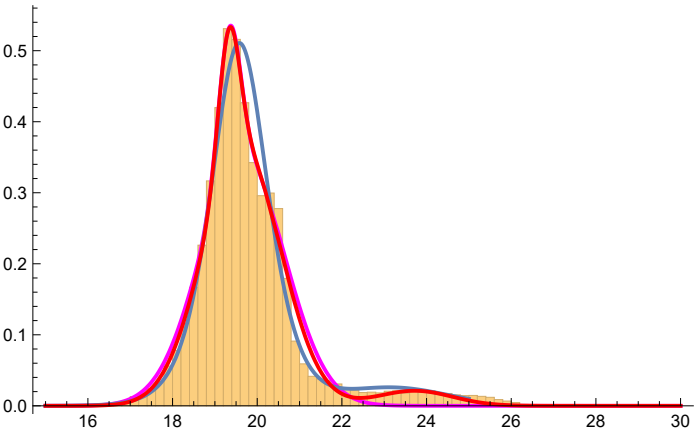
Parallelize: No parallel kernels available; proceeding with sequential evaluation.

```
solOLSu2 = AbsoluteTiming[
           duración absoluta
  NMinimize[mincuad2,
  minimiza aproximadamente
    {{a, 0., 1.}, {b, 0., 1.}, {c, 0, 1}, {u1, 18., 20.}, {s1, 0., 2.}, {u2, 19., 26.},
     {s2, 0., 1.8}, {u3, 19., 26.}, {s3, 0., 4}}, Method → "SimulatedAnnealing"]]
                                                    método
```

Out[•]= {71.3675, {18.8176, {a → 0.217805, b → 1.52786, c → 0.0862539, u1 → 19.3344, s1 → 0.25014, u2 → 19.6398, s2 → 0.922494, u3 → 23.7197, s3 → 0.888782}}}

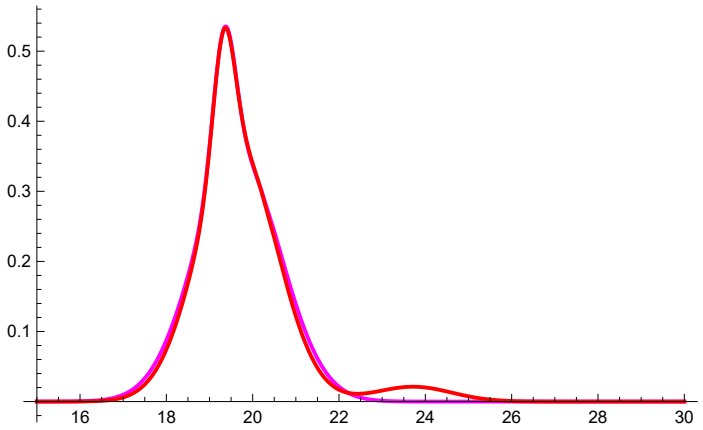
```
Show[
muestra
Histogram[ufilter, {15, 30, 0.2}, "PDF"],
histograma                        función de densidad de probabilidad
Plot[PDF[umodel /. solOLSu[[2, 2]], x], {x, 15, 30}, PlotStyle → Magenta, PlotRange → All],
repr··· función de densidad de probabilidad          estilo de repr··· magenta  rango de rep··· todo
Plot[PDF[bestdistu, x], {x, 15, 25}, PlotRange → All],
repr··· función de densidad de probabilidad  rango de rep··· todo
 Plot[PDF[umodel2 /. solOLSu2[[2, 2]], x], {x, 15, 30}, PlotStyle → Red, PlotRange → All]
  repr··· función de densidad de probabilidad          estilo de repr··· rojo  rango de rep··· todo
]
```

Out[•]=



```
Show[Plot[PDF[umodel /. solOLSu[[2, 2]], x],
 {x, 15, 30}, PlotStyle → Magenta, PlotRange → All],
 Plot[PDF[umodel2 /. solOLSu2[[2, 2]], x], {x, 15, 30}, PlotStyle → Red, PlotRange → All]
]
```

Out[•]=



Como ejercicio propuesto: Realizar los mismos procedimientos para las bandas r_ y g_, iniciar con distribuciones bi-modales y, si su computador lo permite sin llegar a 90 °C, intentar una trimodal. Intentar usar Maximum Likelihood como visto en clases (puede usar la función LogLikelihood[] para obtener la función a maximizar, sea consciente de las limitaciones de su ordenador, intente usar Paralellize cuando sea posible).

Ejercicio 2: 2 set de datos / correlación.

Ahora, se intenta ajustar una función a dos variables (esperablemente) dependientes. En este caso, comprobar la Tercera Ley de Kepler ($T^2 \propto D^3$):

In[•]:=

```
exopl = Import["exoplanetas-11-2024.csv"];
```

```
header2 = exopl[[1]]
```

Out[•]= {name, planet_status, mass, mass_error_min, mass_error_max, mass_sini, mass_sini_error_min, mass_sini_error_max, radius, radius_error_min, radius_error_max, orbital_period, orbital_period_error_min, orbital_period_error_max, semi_major_axis, semi_major_axis_error_min, semi_major_axis_error_max, eccentricity, eccentricity_error_min, eccentricity_error_max, inclination, inclination_error_min, inclination_error_max, angular_distance, discovered, updated, omega, omega_error_min, omega_error_max, tperi, tperi_error_min, tperi_error_max, tconj, tconj_error_min, tconj_error_max, tzero_tr, tzero_tr_error_min, tzero_tr_error_max, tzero_tr_sec, tzero_tr_sec_error_min, tzero_tr_sec_error_max, lambda_angle, lambda_angle_error_min, lambda_angle_error_max, impact_parameter, impact_parameter_error_min, impact_parameter_error_max, tzero_vr, tzero_vr_error_min, tzero_vr_error_max, k, k_error_min, k_error_max, temp_calculated, temp_calculated_error_min, temp_calculated_error_max, temp_measured, hot_point_lon, geometric_albedo, geometric_albedo_error_min, geometric_albedo_error_max, log_g, publication, detection_type, mass_measurement_type, radius_measurement_type, alternate_names, molecules, star_name, ra, dec, mag_v, mag_i, mag_j, mag_h, mag_k, star_distance, star_distance_error_min, star_distance_error_max, star_metallicity, star_metallicity_error_min, star_metallicity_error_max, star_mass, star_mass_error_min, star_mass_error_max, star_radius, star_radius_error_min, star_radius_error_max, star_sp_type, star_age, star_age_error_min, star_age_error_max, star_teff, star_teff_error_min, star_teff_error_max, star_detected_disc, star_magnetic_field, star_alternate_names}

```
Position[header2, "semi_major_axis"]
```
posición
```
Position[header2, "orbital_period"]
```
posición

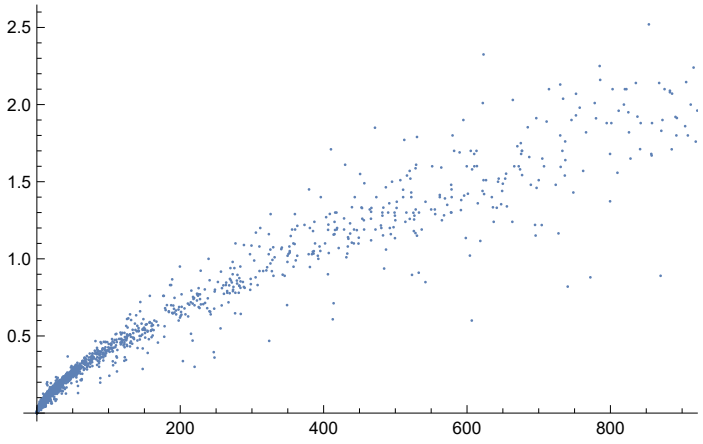
Out[•]= {{15}}

Out[•]= {{12}}

```
In[•]:= orbdist = Select[exopl, NumberQ[#[[12]]] && NumberQ[#[[15]]] &][[All, {12, 15}]];
```
selecciona ¿número? ¿número? todo

```
orbDvsT = ListPlot[orbdist]
          representación de lista
```

Out[•]=




```
nlf = Fit[Log@orbdist, {x, 1}, x]
      aju··  logaritmo
```

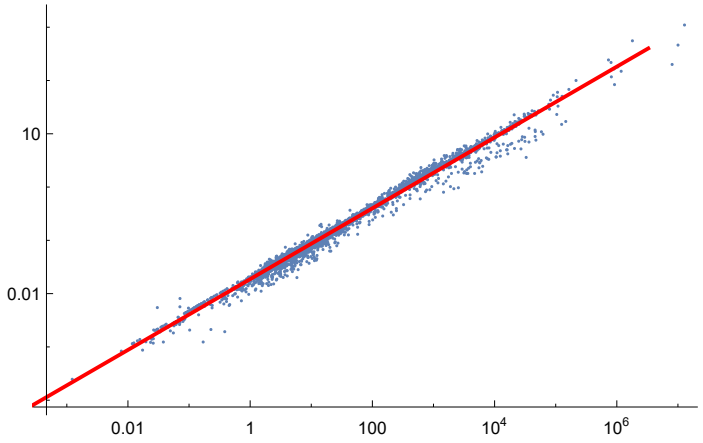
Out[•]= $-3.97849 + 0.664651\,x$

```
nlf2 = NonlinearModelFit[orbdist, a x^n, {n, a}, x]
       ajusta a modelo no lineal
```

Out[•]= $\text{FittedModel}\left[0.0369\,x^{0.607}\right]$

```
Show[
 muestra
 ListLogLogPlot[orbdist],
 representación log log de lista
 Plot[nlf, {x, -10, 15}, PlotStyle → Red]]
 representación gráfica        estilo de repr···  rojo
```

Out[•]=




```
nlf2 // Normal
        normal
```

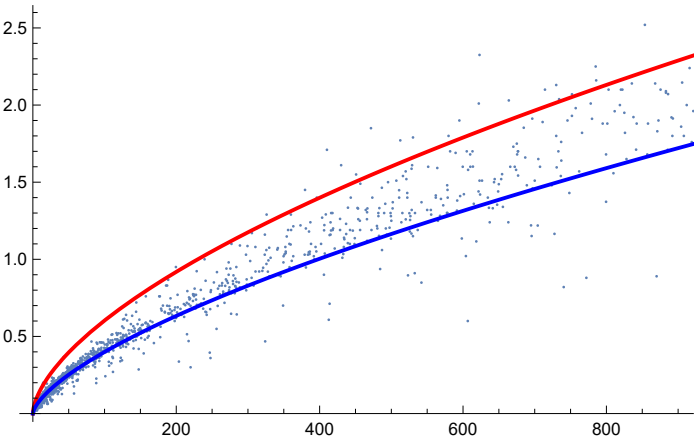
Out[•]= $0.0368935\,x^{0.60679}$

```
nlf1 = E^nlf /. x → Log@x
```
número e logaritm

Out[•]= $0.0187139\, x^{0.664651}$

```
Show[
```
muestra
```
ListPlot[orbdist],
```
representación de lista
```
Plot[nlf2 // Normal, {x, 0, 1000}, PlotStyle → Red],
```
representación··· normal estilo de repr··· rojo
```
Plot[nlf1, {x, 0, 1000}, PlotStyle → Blue]
```
representación gráfica estilo de repr··· azul
```
]
```

Out[•]=




```
end = Now;
```
ahora
```
timeelapsed = end - beginning
```

Out[•]= 6.55128 min