KDE: Ejemplo "V sini"

```
In[ ]:= SetDirectory[NotebookDirectory[]]
```
establece direct··· directorio de cuaderno

```
Out[ ]= /Users/michel/Library/CloudStorage/OneDrive-uv.cl/cursos/pregrado/Estadisticas/2024/
          Clases
```

```
In[ ]:= vsini = Import["vsini.csv"] // Flatten;
```
importa aplana

```
In[ ]:= Length@vsini
```
longitud

```
Out[ ]= 11818
```

```
In[ ]:= Select[vsini, ! NumberQ]
```
selecciona ¿número?

```
Out[ ]= { }
```

```
In[ ]:= Min@vsini
```
mínimo

```
Out[ ]= 0
```

```
In[ ]:= Max@vsini
```
máximo

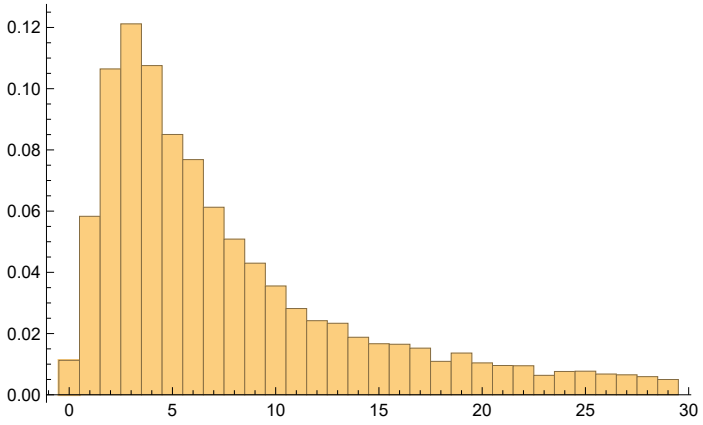
```
Out[ ]= 29
```

```
In[ ]:= Count[vsini, 0]
```
conteo

```
Out[ ]= 133
```

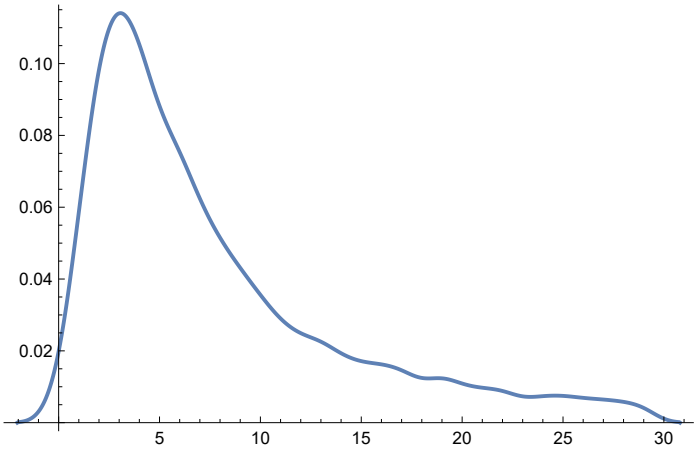
```
In[ ]:= Histogram[vsini, Automatic, "PDF"]
```
histograma automático función

```
Out[ ]=
```



In[ ]:=
```
SmoothHistogram@vsini
```
histograma suave

Out[ ]=

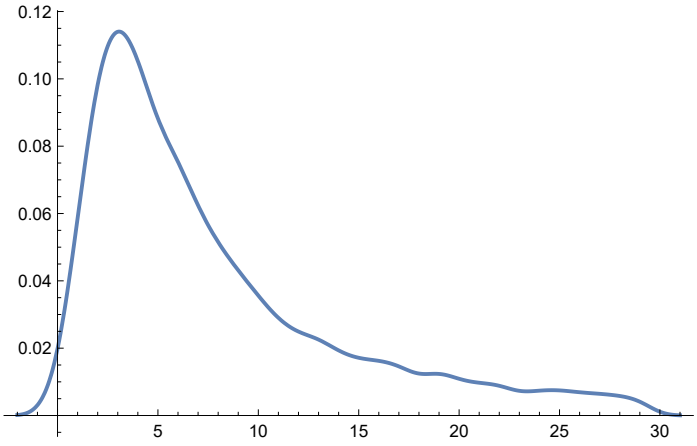


In[ ]:=
```
dist0 = SmoothKernelDistribution[vsini]
```
distribución de núcleo suave

Out[ ]=
```
DataDistribution[
```
Type: SmoothKernel
Data points: 11818
```
]
```

In[ ]:=
```
grd0 = Plot[PDF[dist0, x], {x, -2, 31}]
```
repr··· función de densidad de probabilidad

Out[ ]=



In[ ]:=
```
CDF[dist0, 0]
```
función de distribución acumulada

Out[ ]=
```
0.0107542
```

In[ ]:=
```
NIntegrate[PDF[dist0, x], {x, -∞, 0}]
```
integra numé·· función de densidad de probabilidad

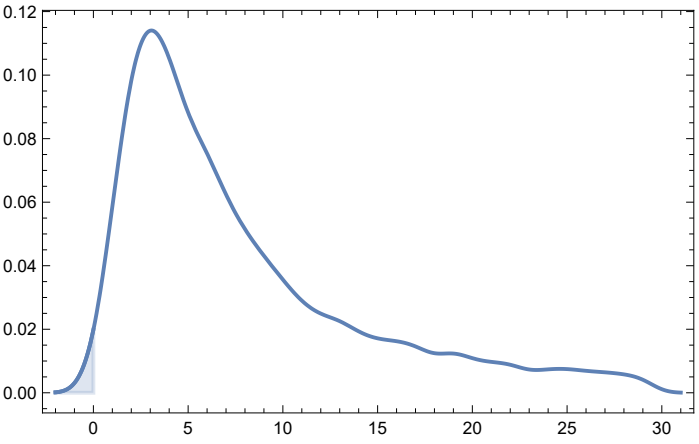
Out[ ]=
```
0.0107542
```

In[ ]:=
```
Probability[x < 0, x ≈ dist0]
```
probabilidad

Out[ ]=
```
0.0107542
```

In[•]:=
```
Show[Plot[PDF[dist0, x], {x, -2, 31}],
 Plot[PDF[dist0, x], {x, -2, 0}, Filling → Bottom], Frame → True]
```
mue·· repr··· función de densidad de probabilidad
repr··· función de densidad de probabilidad relleno abajo marco verdade

Out[•]=



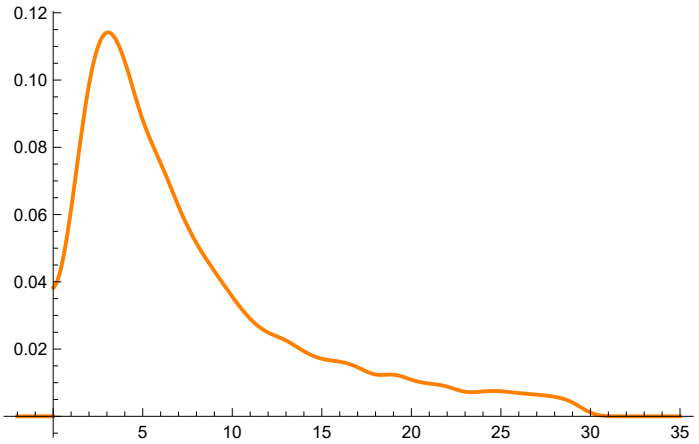
In[•]:=
```
Probability[x ≥ 0, x ≈ dist0]
```
probabilidad

Out[•]=
```
0.989246
```

In[•]:=
```
dist1 = SmoothKernelDistribution[vsini, Automatic, {"Bounded", {0, 35}, "Gaussian"}];
```
distribución de núcleo suave automático

In[•]:=
```
grd1 = Plot[PDF[dist1, x], {x, -2, 35}, PlotStyle → Orange]
```
repr··· función de densidad de probabilidad estilo de repr··· naranja

Out[•]=



In[•]:=
```
Probability[x < 0, x ≈ dist1]
```
probabilidad

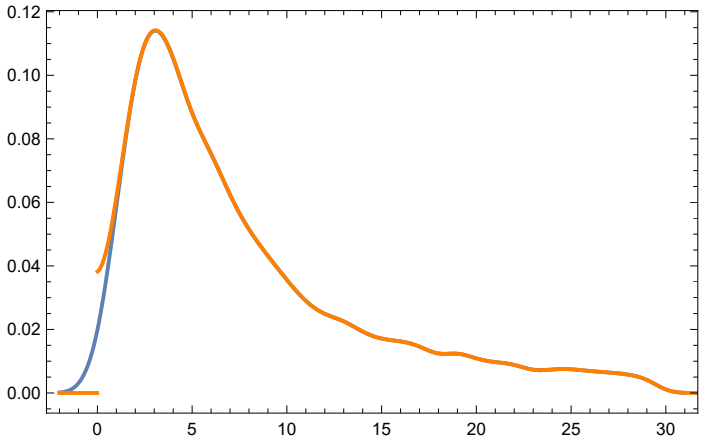
Out[•]=
```
0
```

In[•]:=
```
Probability[x > 0, x ≈ dist1]
```
probabilidad

Out[•]=
```
1.
```

In[ ]:=
```
Show[grd0, grd1, Frame → True]
```
muestra marco verdade

Out[ ]=



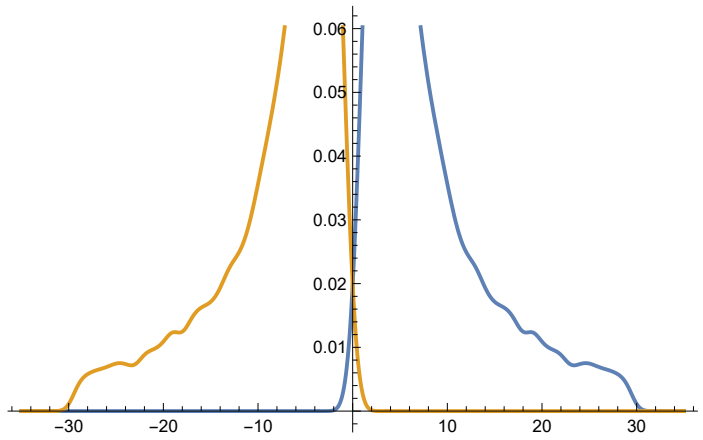
In[ ]:=
```
vsini2 = -vsini;
```

In[ ]:=
```
dist2 = SmoothKernelDistribution[vsini, Automatic, "Gaussian"];
```
distribución de núcleo suave automático

In[ ]:=
```
dist3 = SmoothKernelDistribution[vsini2, Automatic, "Gaussian"];
```
distribución de núcleo suave automático

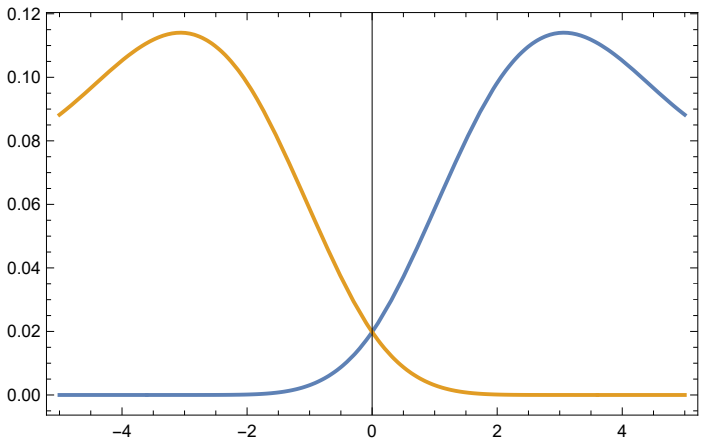
In[ ]:=
```
Plot[{PDF[dist2, x], PDF[dist3, x]}, {x, -35, 35}]
```
repre··· función de densi··· función de densidad de probabilidad

Out[ ]=



In[ ]:=
```
Plot[{PDF[dist2, x], PDF[dist3, x]}, {x, -5, 5}, Frame → True]
```
repre··· función de densi··· función de densidad de probabilidad marco verdade

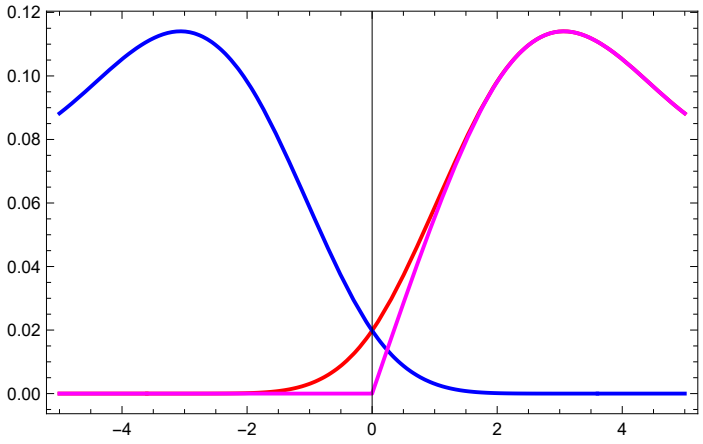
Out[ ]=



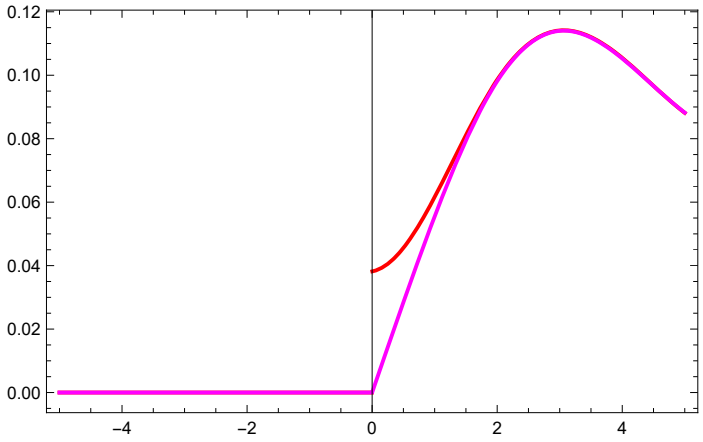
In[ ]:=
```
dist4 = ProbabilityDistribution[PDF[dist2, x] - PDF[dist3, x], {x, 0, 35}];
```
distribución de probabilidad función de densid··· función de densidad de probabilidad

In[ ]:=
```
Plot[{PDF[dist2, x], PDF[dist3, x], PDF[dist4, x]},
 {x, -5, 5}, PlotStyle → {Red, Blue, Magenta}, Frame → True]
```
repre··· función de densi··· función de densi··· función de densidad de probabili
estilo de repre··· rojo azul magenta marco verdade

Out[ ]=



In[ ]:=
```
Plot[{PDF[dist1, x], PDF[dist4, x]}, {x, -5, 5}, PlotStyle → {Red, Magenta}, Frame → True]
```
repre··· | función de densi··· | función de densidad de probabilidad | estilo de repre··· | rojo | magenta | marco | verdade

Out[ ]=


In[ ]:=
```
cte = NIntegrate[PDF[dist4, x], {x, 0, 50}]
```
integra numé·· | función de densidad de probabilidad

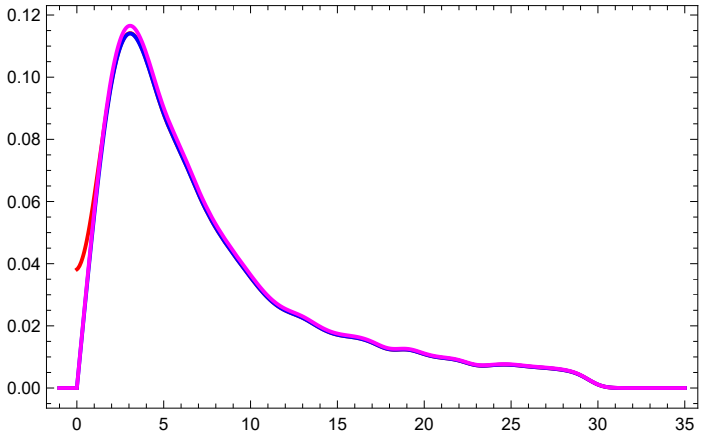
Out[ ]=
```
0.978492
```

In[ ]:=
```
dist5 = ProbabilityDistribution[PDF[dist4, x] / cte, {x, 0, 35}];
```
distribución de probabilidad | función de densidad de probabilidad

In[ ]:=
```
NIntegrate[PDF[dist5, x], {x, 0, 50}]
```
integra numé·· | función de densidad de probabilidad

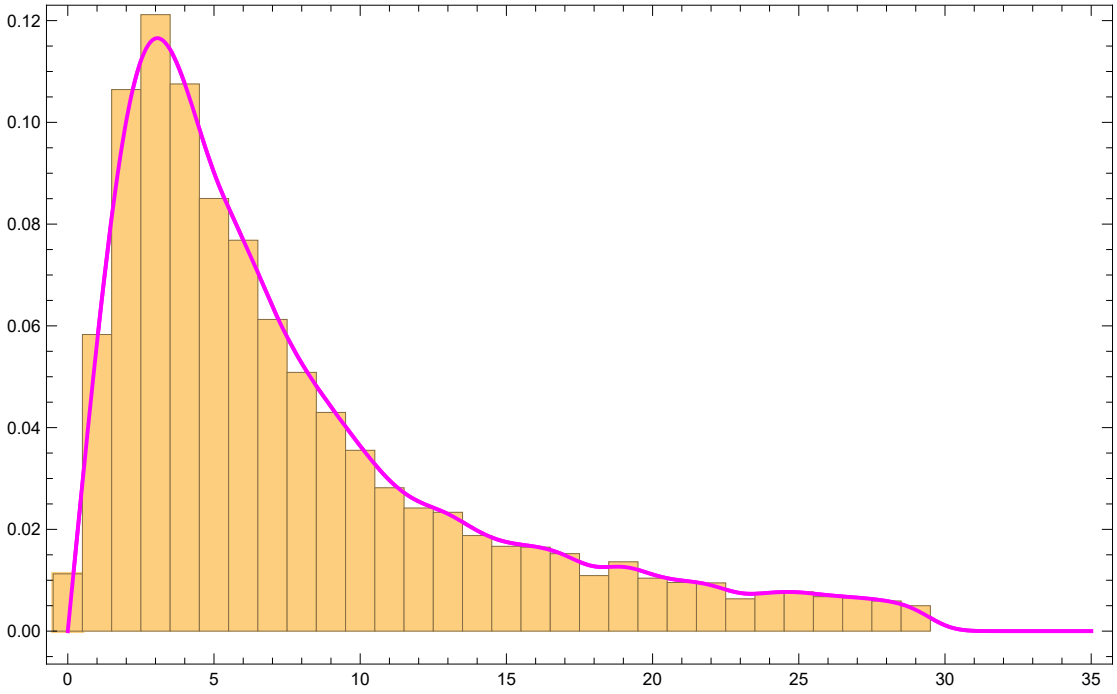
Out[ ]=
```
1.
```

In[ ]:=
```
Plot[{PDF[dist1, x], PDF[dist4, x], PDF[dist5, x]},
 {x, -1, 35}, PlotStyle → {Red, Blue, Magenta}, Frame → True]
```
repre··· | función de densi··· | función de densi··· | función de densidad de probabilida
estilo de repre··· | rojo | azul | magenta | marco | verdade

Out[ ]=


In[ ]:=
```
Show[Plot[PDF[dist5, x], {x, 0, 35}, PlotStyle → Magenta],
 Histogram[vsini, Automatic, "PDF"],
 Plot[PDF[dist5, x], {x, 0, 35}, PlotStyle → Magenta], Frame → True, ImageSize → Large]
```

Out[ ]=



Ejercicio : de la seccion "A rule-of-thumb bandwidth estimator" de la pagina web "https://en.wikipedia.org/wiki/Kernel_density_estimation"

A partir de que distribución provienen los datos de "V sini"

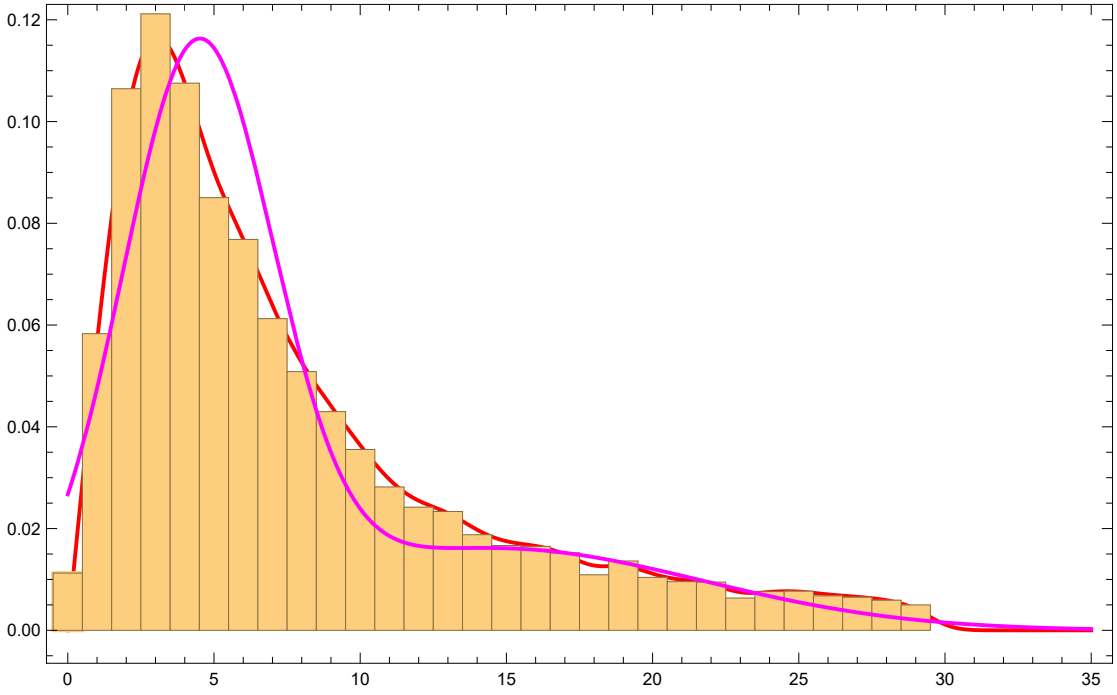
Minimos Cuadrados

In[ ]:=
```
dist2N = FindDistribution[vsini, TargetFunctions → "Continuous"]
```

Out[ ]=
```
MixtureDistribution[{0.712421, 0.287579},
 {NormalDistribution[4.45085, 2.57368], NormalDistribution[14.5821, 7.11049]}]
```

In[•]:=
```
Show[Plot[PDF[dist5, x], {x, 0, 35}, PlotStyle → Red], Histogram[vsini, Automatic, "PDF"],
 Plot[PDF[dist2N, x], {x, 0, 35}, PlotStyle → Magenta], Frame → True, ImageSize → Large]
```
mue·· repr··· función de densidad de probabili··· estilo de repr··· rojo histograma automático función d
repr··· función de densidad de probabilidad estilo de repr··· magenta marco verd··· tamaño de i··· grande

Out[•]=



In[•]:=
```
modelo = MixtureDistribution[{a1, a2},
 {NormalDistribution[μ1, σ1], NormalDistribution[μ2, σ2]}]
```
distribución mezcla
distribución normal distribución normal

Out[•]=
```
MixtureDistribution[{a1, a2}, {NormalDistribution[μ1, σ1], NormalDistribution[μ2, σ2]}]
```

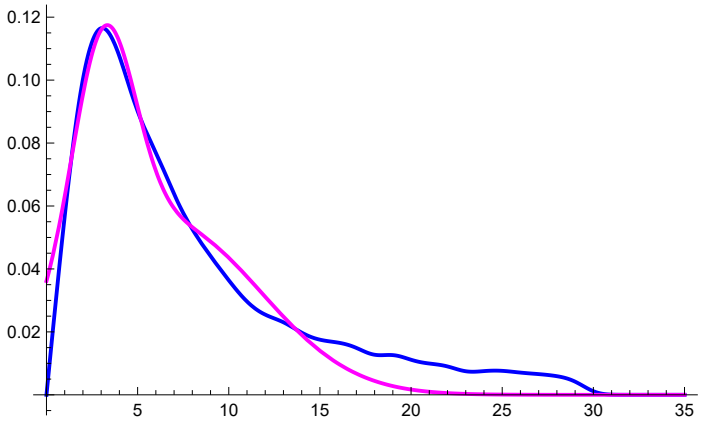
In[•]:=
```
mincuad =
 ParallelSum[(PDF[dist5, vsini[[k]]] - PDF[modelo, vsini[[k]]])^2, {k, 1, Length@vsini}];
```
suma en paralelo función de densidad de pro··· función de densidad de probabilidad longitud

In[•]:=
```
solMC = NMinimize[mincuad, {{a1, 0.1, .9}, {a2, 0.1, .9}, {μ1, 4., 5.},
 {σ1, 2., 3.}, {μ2, 12., 17.}, {σ2, 5., 9.}}, Method → "SimulatedAnnealing"]
```
minimiza aproximadamente
método

Out[•]=
```
{0.470675,
 {a1 → 0.507909, a2 → 1.061, μ1 → 3.11463, σ1 → 1.70988, μ2 → 6.75482, σ2 → 5.03249}}
```

```
Plot[{PDF[dist5, x], PDF[modelo, x] /. solMC[[2]]}, {x, 0, 35}, PlotStyle → {Blue, Magenta}]
```
repre··· función de densi··· función de densidad de probabilidad estilo de repre··· azul magenta

Out[•]=



Maxima verosimilitud

In[•]:=
```
modelo = MixtureDistribution[{a1, a2},
```
distribución mezcla
```
  {NormalDistribution[μ1, σ1], NormalDistribution[μ2, σ2]}]
```
distribución normal distribución normal

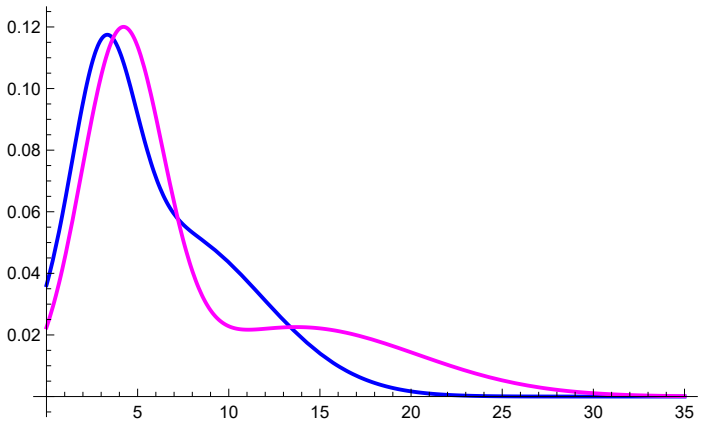
In[•]:=
```
loglik = LogLikelihood[modelo, vsini];
```
verosimilitud logarítmica

In[•]:=
```
solML = NMaximize[{Re@loglik, a1 > 0, a2 > 0}, {{a1, 0.1, .9}, {a2, 0.1, .9}, {μ1, 4., 5.},
```
maximiza ap··· parte real
```
   {σ1, 2., 3.}, {μ2, 12., 17.}, {σ2, 5., 9.}}, Method → "SimulatedAnnealing"]
```
método

Out[•]=
```
{-35 690.2,
 {a1 → 1.57467, a2 → 0.937125, μ1 → 4.1516, σ1 → 2.23115, μ2 → 13.7308, σ2 → 6.60134}}
```

In[•]:=
```
Plot[{PDF[modelo, x] /. solMC[[2]], PDF[modelo, x] /. solML[[2]]},
```
repre··· función de densidad de probabilidad función de densidad de probabilidad
```
 {x, 0, 35}, PlotStyle → {Blue, Magenta}]
```
estilo de repre··· azul magenta

Out[•]=



```
In[ ]:= solML = NMaximize[{Re@loglik, a1 > 0, a2 > 0}, {{a1, 0.1, .9}, {a2, 0.1, .9},
                maximiza ap··· | parte real
          {μ1, 4., 5.}, {σ1, 2., 3.}, {μ2, 12., 17.}, {σ2, 5., 9.}}, Method → "NelderMead"]
                                                                    método
Out[ ]= {-35 690.2,
         {a1 → 0.57266, a2 → 0.340804, μ1 → 4.1516, σ1 → 2.23115, μ2 → 13.7308, σ2 → 6.60134}}

In[ ]:= solML = NMaximize[{Re@loglik, a1 > 0, a2 > 0}, {{a1, 0.1, .9}, {a2, 0.1, .9}, {μ1, 4., 5.},
                maximiza ap··· | parte real
          {σ1, 2., 3.}, {μ2, 12., 17.}, {σ2, 5., 9.}}, Method → "DifferentialEvolution"]
                                                                    método

   ••• NMaximize: The function value Indeterminate is not a number at {a1, a2, μ1, μ2, σ1, σ2} =
       {0., 0., 4.2913, 13.8269, 2.24668, 6.53956}.

Out[ ]= {-35 691.,
         {a1 → 1.01073, a2 → 0.585392, μ1 → 4.13945, σ1 → 2.22881, μ2 → 13.8353, σ2 → 6.57494}}

In[ ]:= solML = NMaximize[{Re@loglik, a1 > 0, a2 > 0, σ1 > 0, σ2 > 0}, {{a1, 0.1, .9}, {a2, 0.1, .9},
                maximiza ap··· | parte real
          {μ1, 4., 5.}, {σ1, 2., 3.}, {μ2, 12., 17.}, {σ2, 5., 9.}}, Method → "RandomSearch"]
                                                                    método
Out[ ]= {-35 690.2,
         {a1 → 0.76704, a2 → 0.456484, μ1 → 4.1516, σ1 → 2.23115, μ2 → 13.7308, σ2 → 6.60134}}
```