

Blatt 06 - Praktische Optimierung - Adrian Lentz, Robert

Lösungen und Erklärungen für Blatt 06.

\newline

Adrian Lentz - Matrikelnummer: 258882

\newline

Robert Schönewald - Matrikelnummer: 188252

Aufgabe 6.1

In Kapitel 6 auf Folien 40 bis 42 haben wir bereits gesehen, wie die Maximum Entropie Verteilungen bestimmt werden. Das Schlüsselement ist hierbei die Bestimmung von q , welches von v abhängig ist. Dazu betrachten wir folgende Funktion:

```
In [1]: import numpy as np
import timeit
import matplotlib.pyplot as plt
import scipy
import statsmodels.api as sm
import statsmodels.distributions.empirical_distribution as edf
from scipy.stats import multivariate_normal
```

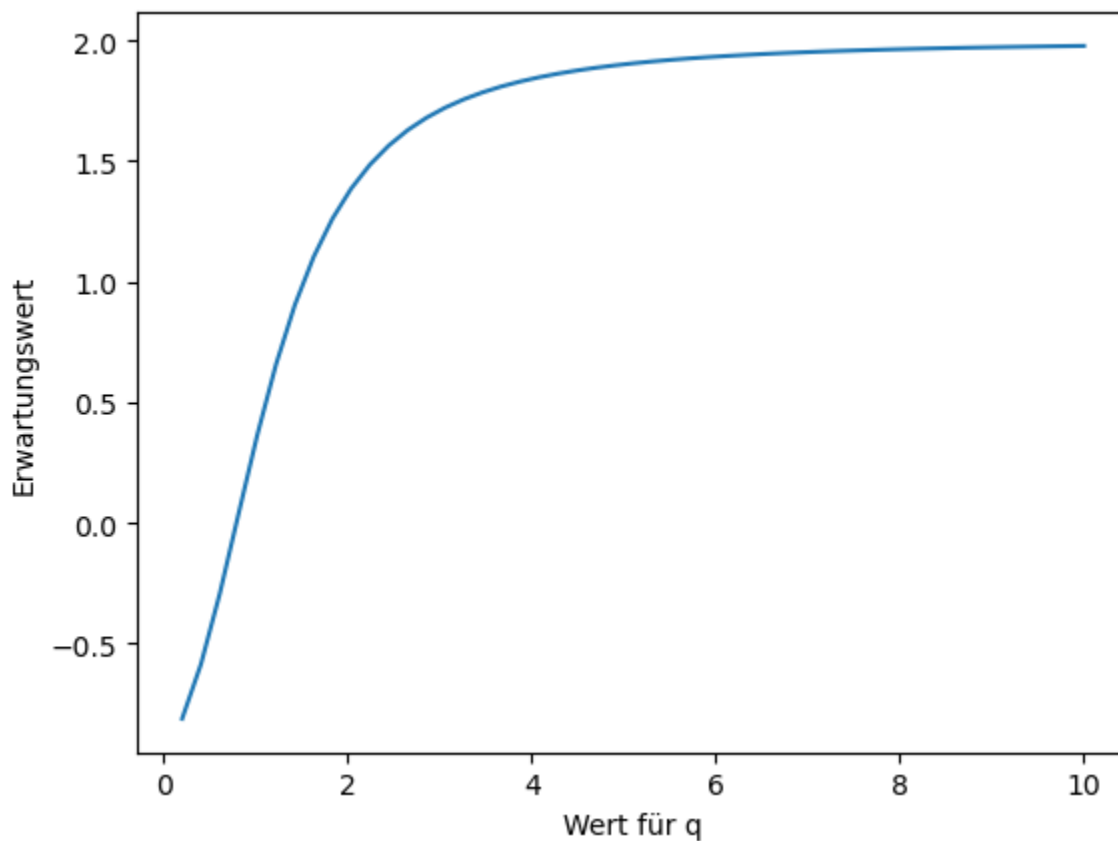
```
In [2]: #Funktion definieren
def f(x):
    return (-x**(-1)+2*x**2) / ( (x**-(1)) + 1 + (x**2) ) #Folien der Vorlesung auf konkre
```

```
In [3]: x=np.linspace(0,10)
#Graph für Funktion f_1(x)

plt.xlabel("Wert für q")
plt.ylabel("Erwartungswert")
plt.plot(x, f(x));

plt.show()
```

```
C:\Users\Robert\AppData\Local\Temp\ipykernel_17444\31228444.py:3: RuntimeWarning: divide
by zero encountered in reciprocal
    return (-x**(-1)+2*x**2) / ( (x**-(1)) + 1 + (x**2) ) #Folien der Vorlesung auf konkret
en Träger angewandt ergibt diese Funktion zur Berechnung von q
C:\Users\Robert\AppData\Local\Temp\ipykernel_17444\31228444.py:3: RuntimeWarning: invali
d value encountered in divide
    return (-x**(-1)+2*x**2) / ( (x**-(1)) + 1 + (x**2) ) #Folien der Vorlesung auf konkret
en Träger angewandt ergibt diese Funktion zur Berechnung von q
```



q darf nicht negativ sein, da sonst Wahrscheinlichkeiten ebenfalls negativ werden. Man erkennt, dass v Werte von 0 bis 2 annehmen kann, wobei 2 nur ein Grenzwert für sehr hohe q-Werte ist.

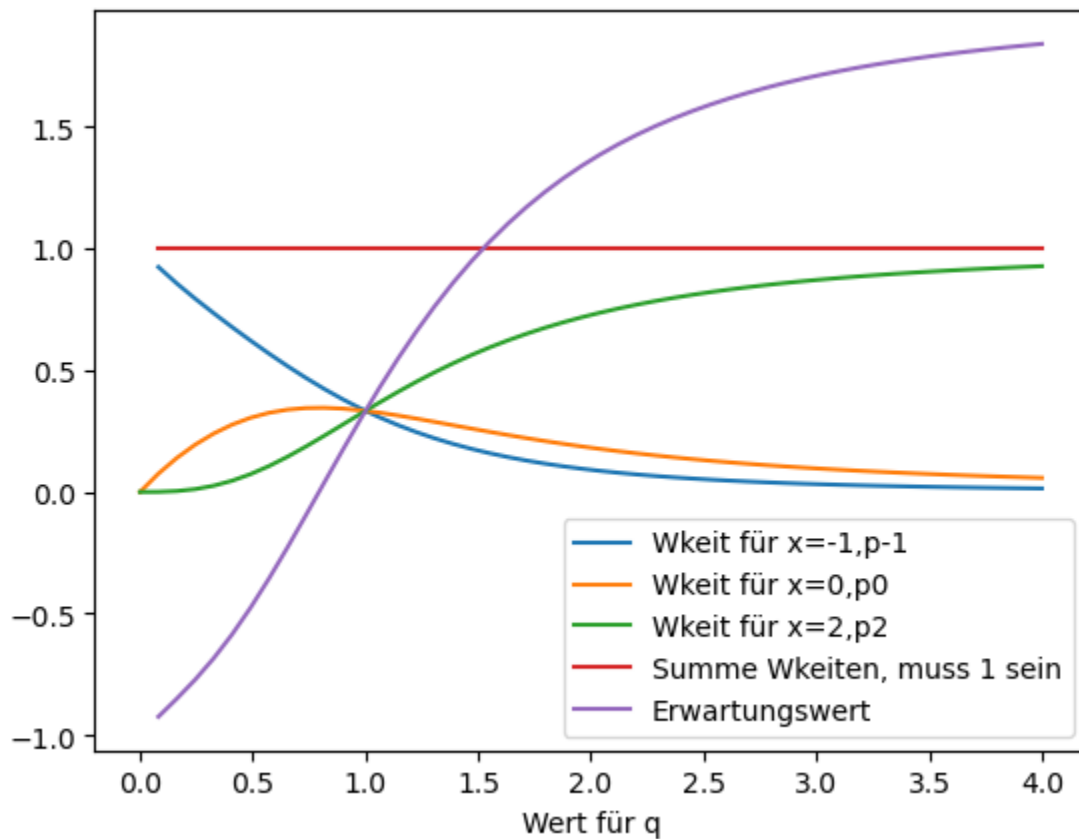
```
In [4]: def f1(x):
        return x**(-1)/(x**(-1)+x**0+x**2)
        def f2(x):
            return x**0/(x**(-1)+x**0+x**2)
        def f3(x):
            return x**2/(x**(-1)+x**0+x**2)
```

```
In [5]: x=np.linspace(0,4)
        #Graph für Funktion f_1(x)

        plt.xlabel("Wert für q")
        plt.plot(x, f1(x),label=r"Wkeit für x=-1,p-1");
        plt.plot(x, f2(x),label=r"Wkeit für x=0,p0");
        plt.plot(x, f3(x),label=r"Wkeit für x=2,p2");
        plt.plot(x, f1(x)+f2(x)+f3(x),label=r"Summe Wkeiten, muss 1 sein");
        plt.plot(x, -f1(x)+2*f3(x),label=r"Erwartungswert");
        plt.legend();

        plt.show()
```

```
C:\Users\Robert\AppData\Local\Temp\ipykernel_17444\1387267780.py:2: RuntimeWarning: divide by zero encountered in reciprocal
    return x**(-1)/(x**(-1)+x**0+x**2)
C:\Users\Robert\AppData\Local\Temp\ipykernel_17444\1387267780.py:2: RuntimeWarning: invalid value encountered in divide
    return x**(-1)/(x**(-1)+x**0+x**2)
C:\Users\Robert\AppData\Local\Temp\ipykernel_17444\1387267780.py:4: RuntimeWarning: divide by zero encountered in reciprocal
    return x**0/(x**(-1)+x**0+x**2)
C:\Users\Robert\AppData\Local\Temp\ipykernel_17444\1387267780.py:6: RuntimeWarning: divide by zero encountered in reciprocal
    return x**2/(x**(-1)+x**0+x**2)
```

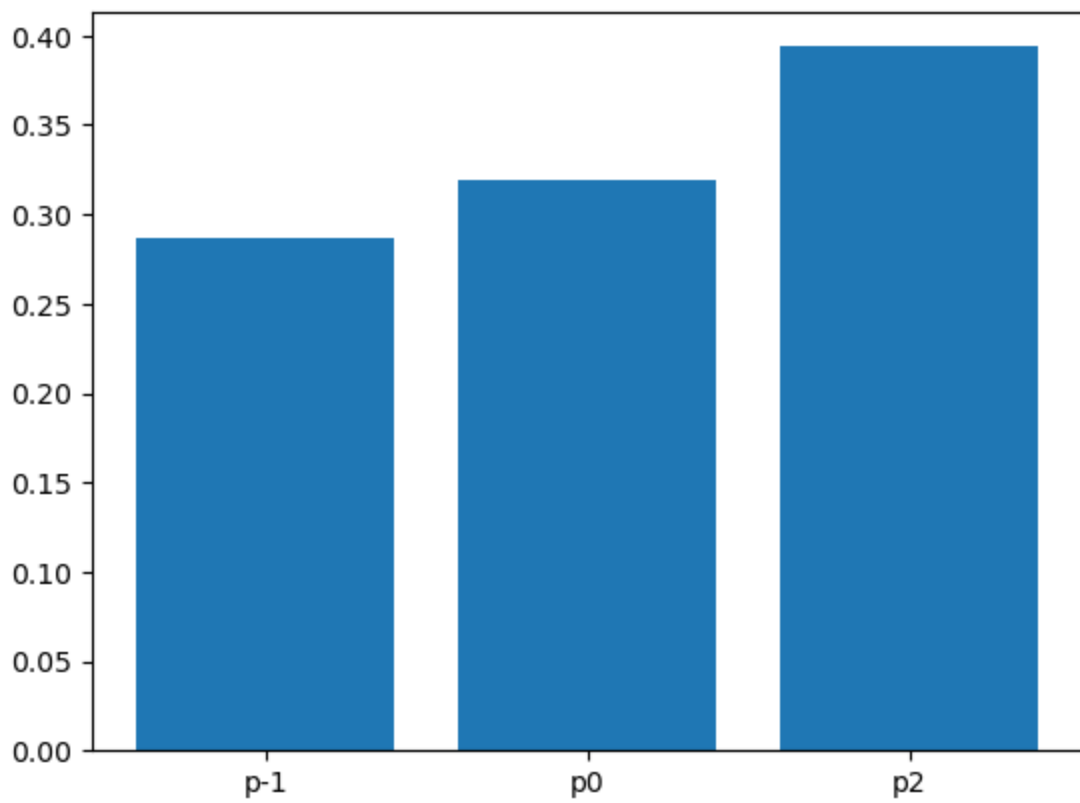


Man erkennt, dass mit höheren Erwartungswerten die Wahrscheinlichkeit für $x=2$ immer höher werden während die anderen kleiner werden und mit niedrigeren Erwartungswerten die Wkeit für $x=-1$ steigt während die anderen Wkeiten sinken. So ein Ergebnis ist auch zu erwarten.

Lege nun $v=0,5$ fest: Daraus folgt für q : $q=10/9$

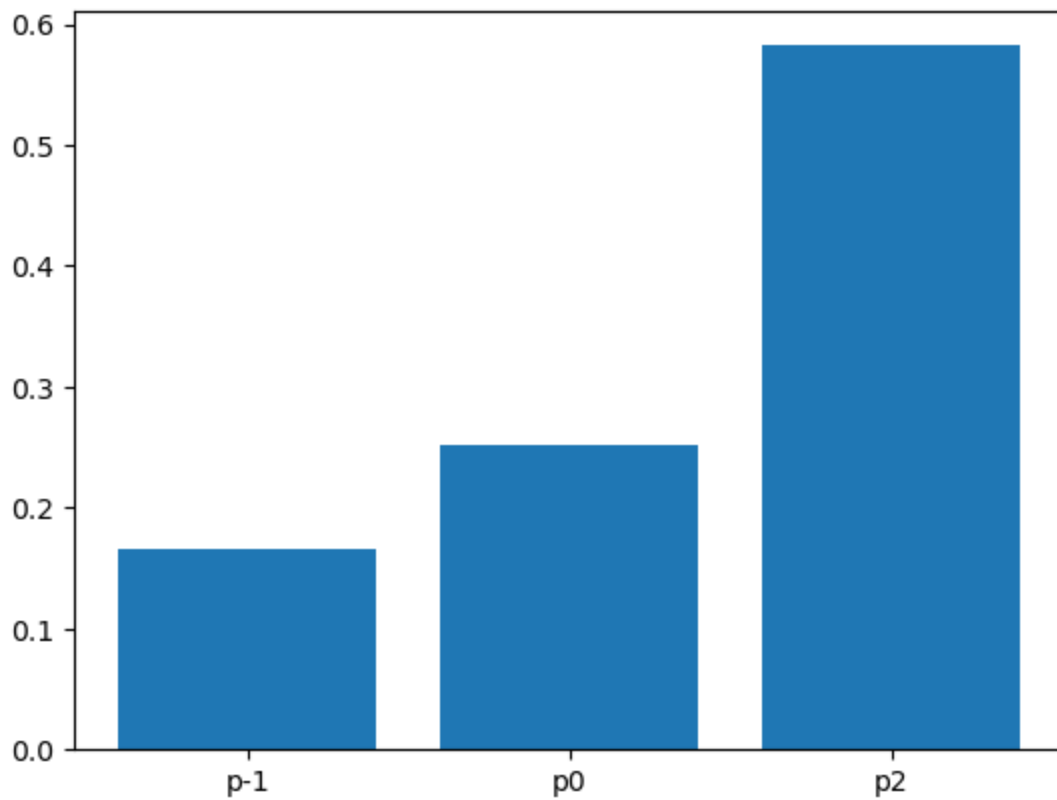
```
In [6]: x=10/9
y=[f1(x), f2(x), f3(x)]

plt.bar(["p-1", "p0", "p2"], y)
plt.show()
```



v=1. Daraus folgt $q=1.52$

```
In [7]: x=1.52  
y=[f1(x), f2(x), f3(x)]  
  
plt.bar(["p-1", "p0", "p2"], y)  
plt.show()
```

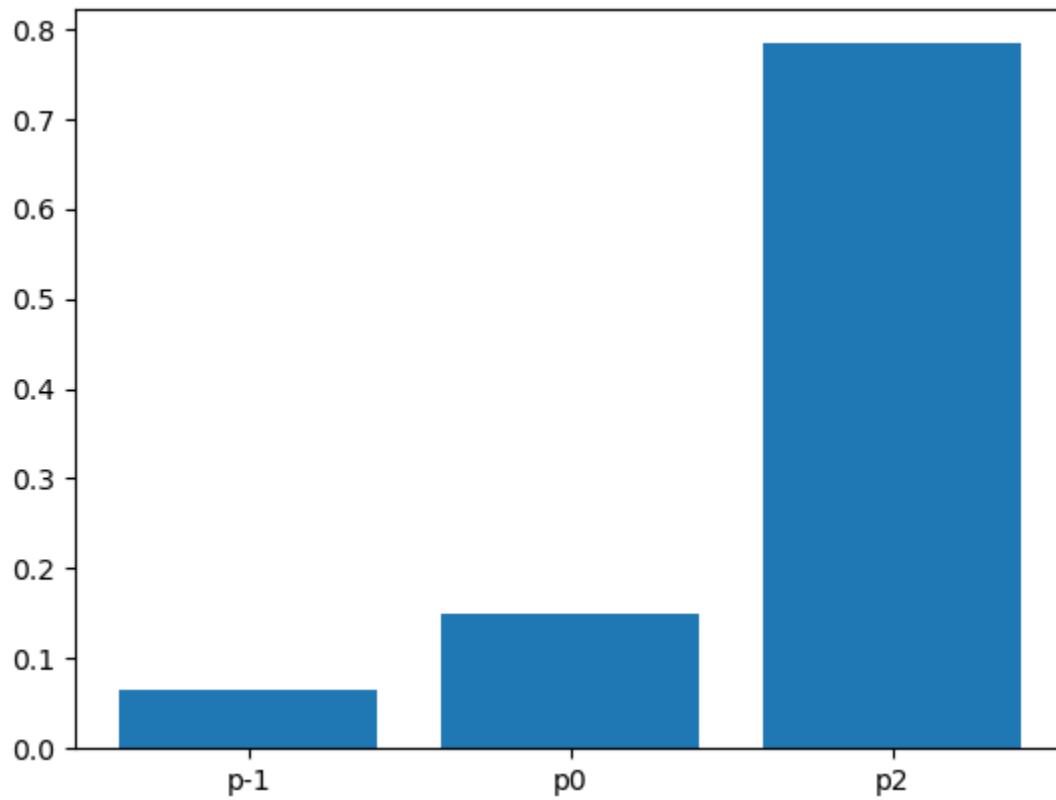


v=1,5. Daraus folgt $q=2.28$

```
In [8]: x=2.28
```

```
y=[f1(x), f2(x), f3(x)]
```

```
plt.bar(["p-1", "p0", "p2"], y)  
plt.show()
```



Ähnlich wie im Diagramm mit den einzelnen Wahrscheinlichkeiten verhalten sich auch die Verteilungen.

In []: