Auswertung

June 20, 2024

1 Auswertung B3.1

[1]:
```julia
using CSV
using DataFrames
using Plots
using LaTeXStrings
using Statistics
```

1.1 Rechnungen zur Vorbereitung

[2]:
```julia
T_12 = 30.08 * 365.25 * 24 * 60 * 60 # s
```

[2]:
```
9.492526079999999e8
```

[3]:
```julia
log(2) / T_12
```

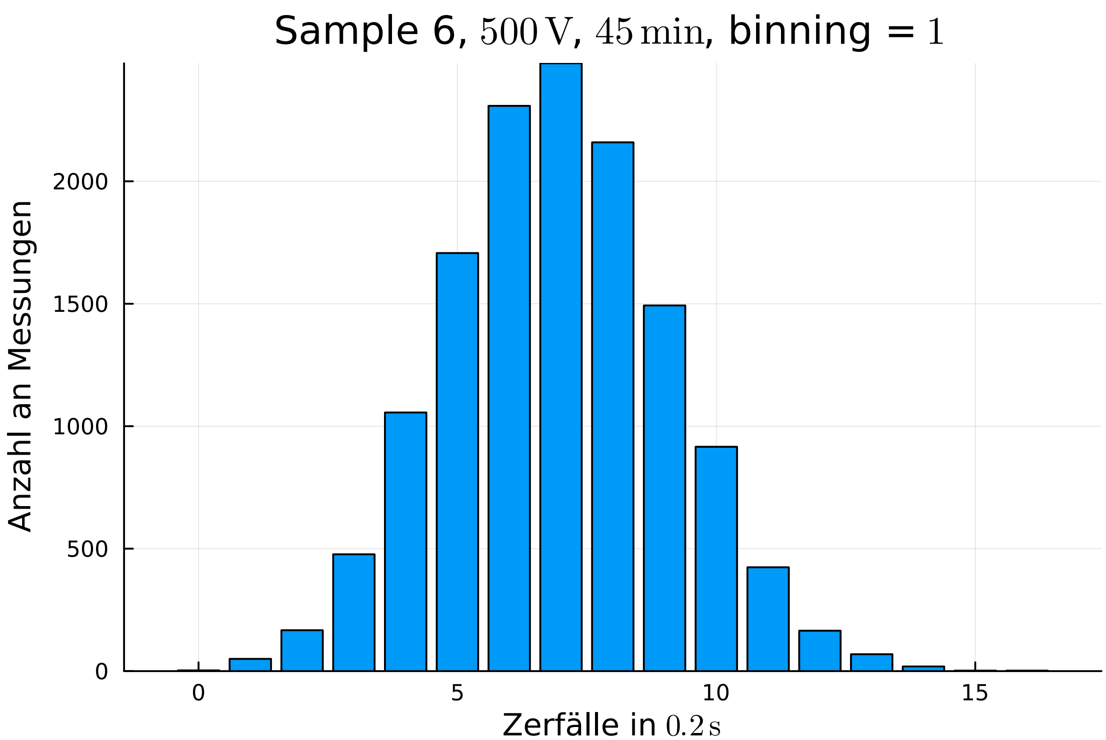
[3]:
```
7.302030826339804e-10
```

1.2 0: Rohe Messdaten

Sample 6, 500 V, 45min:

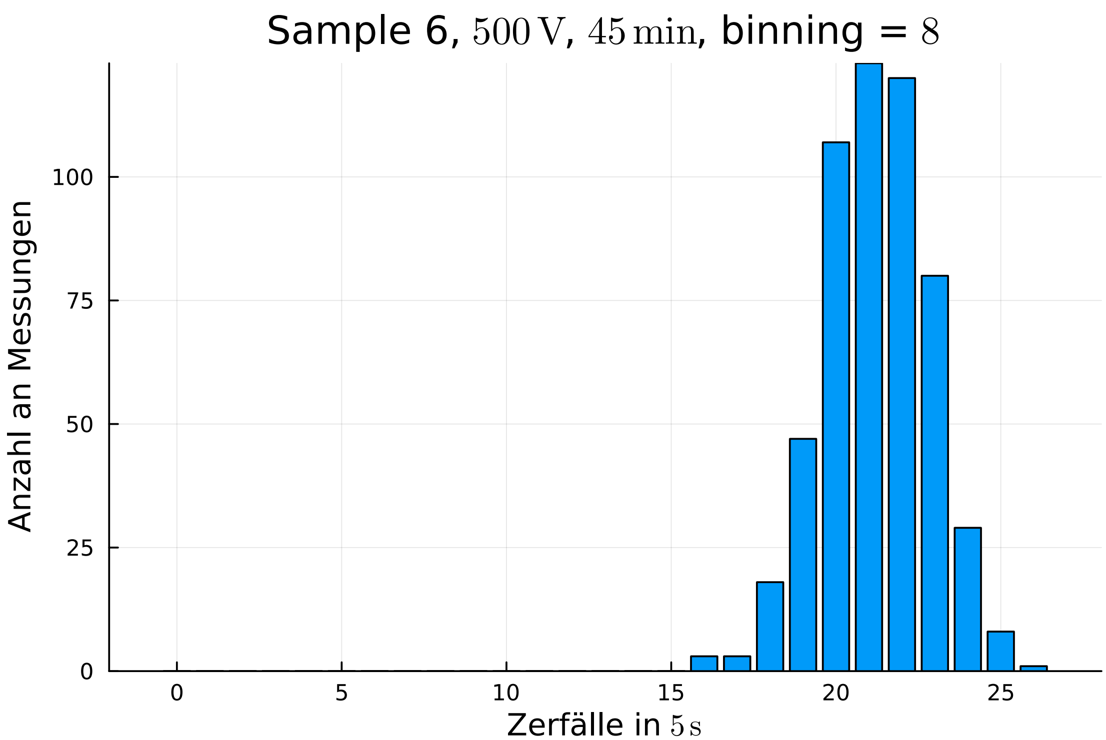
[4]:
```julia
# ΔT = 0.2, binning = 1
zerfälle_6_500_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
anzahl_6_500_1 =␣
 ↪[3,50,167,477,1056,1707,2308,2482,2159,1493,916,424,165,69,19,2,2]
plot(bar(zerfälle_6_500_1,anzahl_6_500_1,label=""), title=L"Sample 6,␣
 ↪$500\mathrm{\,V}$, $45\mathrm{\,min}$, binning = $1$")
xlabel!(L"Zerfälle in $0.2\mathrm{\,s}$")
ylabel!("Anzahl an Messungen")
```

[4]:



[5]:
```
# ΔT = 5s, binning = 8
zerfälle_6_500_2 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26]
anzahl_6_500_2 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,18,47,107,123,120,80,29,8,1]
plot(bar(zerfälle_6_500_2,anzahl_6_500_2,label=""), title=L"Sample 6, $500\mathrm{\,V}$, $45\mathrm{\,min}$, binning = $8$")
xlabel!(L"Zerfälle in $5\mathrm{\,s}$")
ylabel!("Anzahl an Messungen")
```

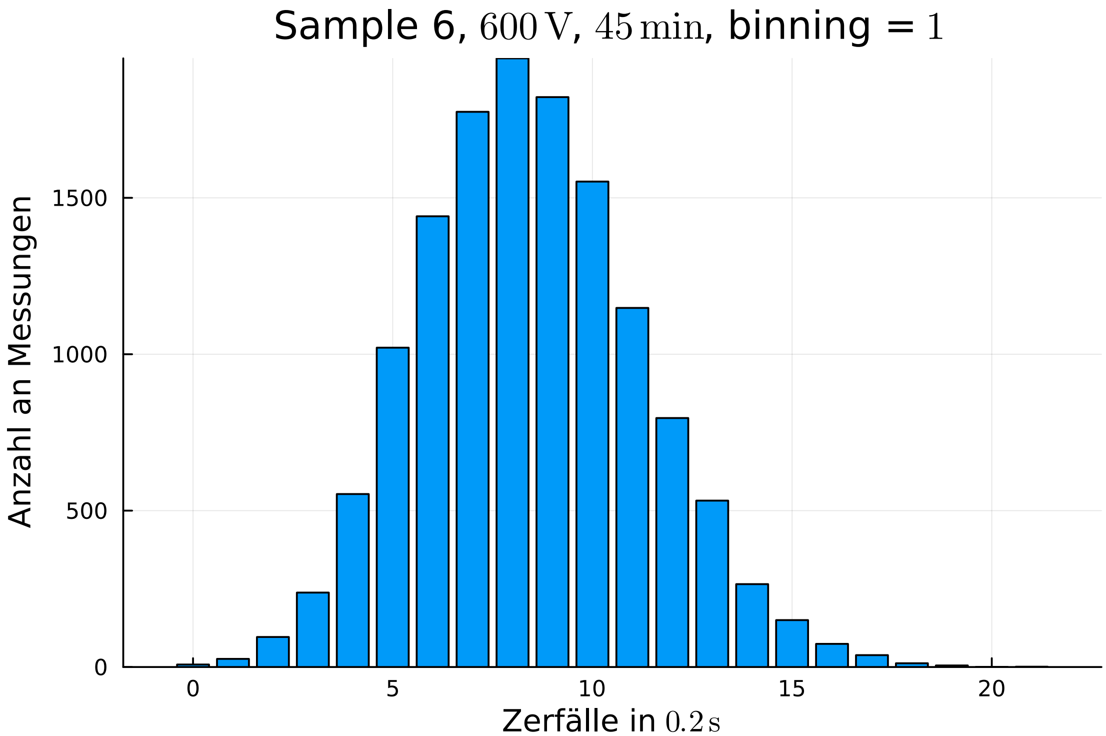
[5]:



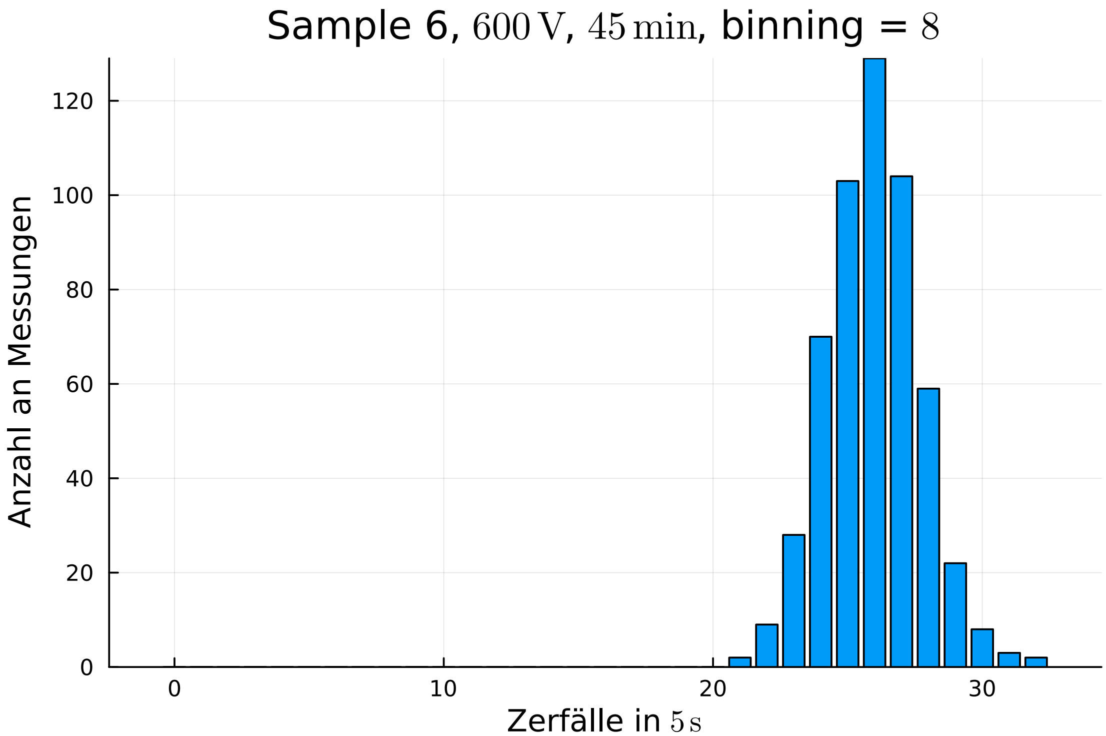
Sample 6, 600V, 45min:

[6]:
```
# ΔT = 0.2s, binning = 1
zerfälle_6_600_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]
anzahl_6_600_1 =␣
  ↪[8,26,96,238,553,1021,1441,1775,1946,1822,1552,1148,796,532,265,150,74,38,12,5,0,1]
plot(bar(zerfälle_6_600_1,anzahl_6_600_1,label=""), title=L"Sample 6,␣
  ↪$600\mathrm{\,V}$, $45\mathrm{\,min}$, binning = $1$")
xlabel!(L"Zerfälle in $0.2\mathrm{\,s}$")
ylabel!("Anzahl an Messungen")
```

[6]:



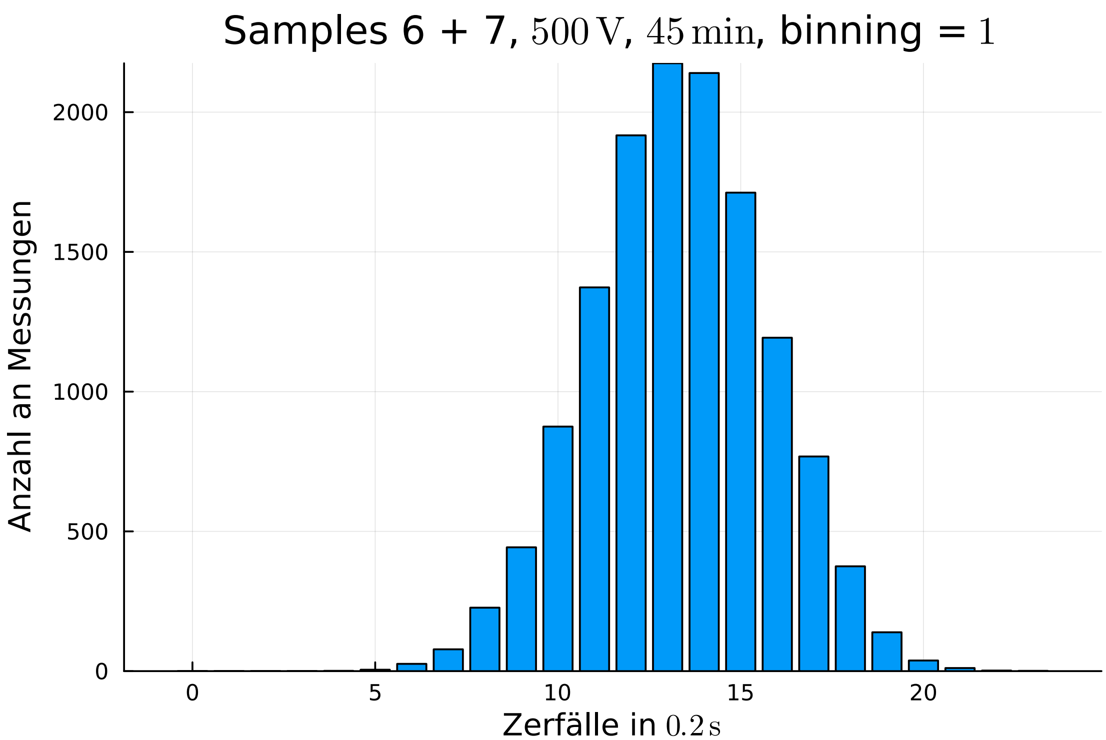
[7]:
```
# ΔT = 5s, binning = 8
zerfälle_6_600_2 =
 ↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32]
anzahl_6_600_2 =
 ↪[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,9,28,70,103,129,104,59,22,8,3,2]
plot(bar(zerfälle_6_600_2,anzahl_6_600_2,label=""), title=L"Sample 6,
 ↪$600\mathrm{\,V}$, $45\mathrm{\,min}$, binning = $8$")
xlabel!(L"Zerfälle in $5\mathrm{\,s}$")
ylabel!("Anzahl an Messungen")
```
[7]:



Sample 6+7, 500V, 45min:

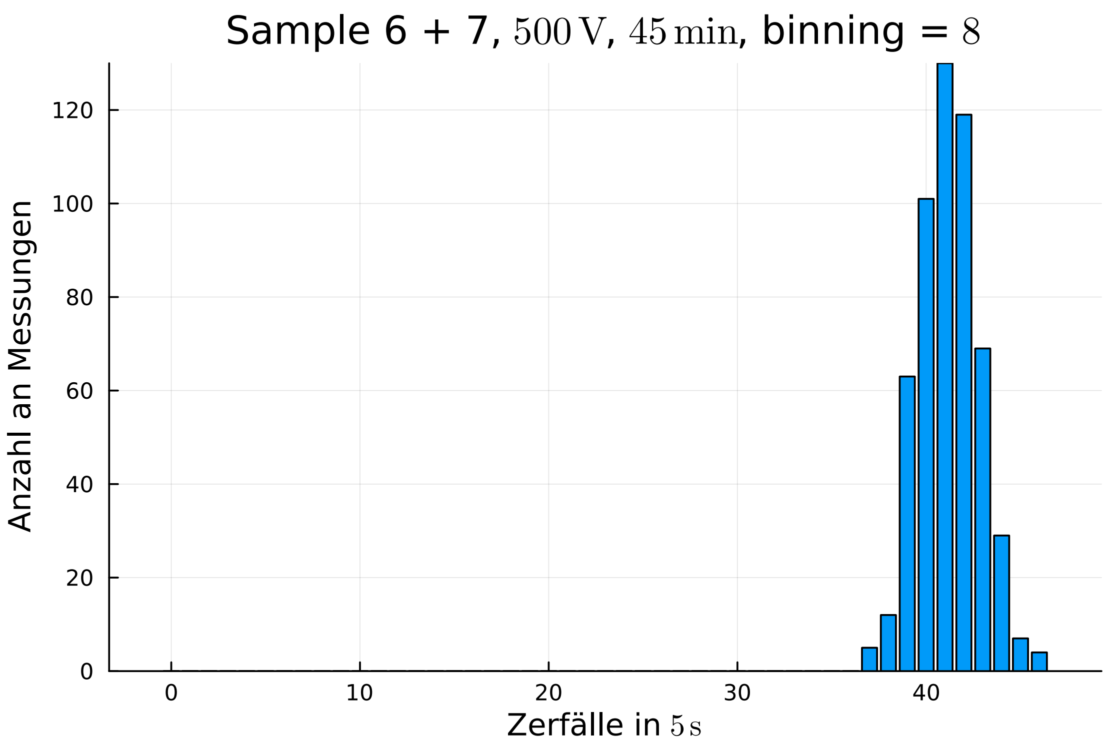
[8]:
```
# ΔT = 0.2s, binning = 1
zerfälle_6und7_500_1 =␣
 ↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]
anzahl_6und7_500_1 =␣
 ↪[0,0,0,0,1,5,26,78,227,443,875,1373,1917,2175,2140,1712,1193,768,375,139,38,11,2,1]
plot(bar(zerfälle_6und7_500_1,anzahl_6und7_500_1,label=""), title=L"Samples 6 +␣
 ↪7, $500\mathrm{\,V}$, $45\mathrm{\,min}$, binning = $1$")
xlabel!(L"Zerfälle in $0.2\mathrm{\,s}$")
ylabel!("Anzahl an Messungen")
```

[8]:




```
[9]: # ΔT = 5s, binning = 8
zerfälle_6und7_500_2 =␣
 ↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,3
    35,36,37,38,39,40,41,42,43,44,45,46]
anzahl_6und7_500_2 =␣
 ↪[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,12,63,101,130,
    29,7,4]
plot(bar(zerfälle_6und7_500_2,anzahl_6und7_500_2,label=""), title=L"Sample 6 +␣
 ↪7, $500\mathrm{\,V}$, $45\mathrm{\,min}$, binning = $8$")
xlabel!(L"Zerfälle in $5\mathrm{\,s}$")
ylabel!("Anzahl an Messungen")
```

[9]:



1.3 1: Poisson-Verteilung

[10]:
```
# Poissonverteilung:
lambda = 10 # (Testwert) Mittelwert lambda
N = 10 # (Testwert) Normierungsfaktor N = Summe der Höhe aller Balken
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))
```

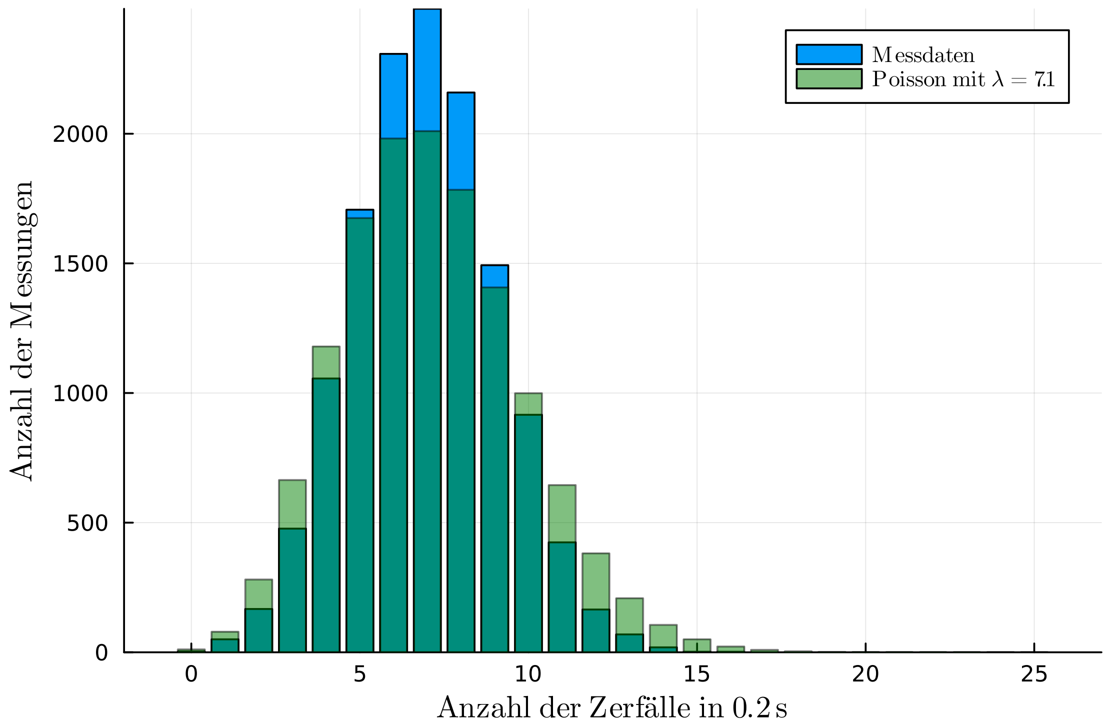
[10]: P (generic function with 1 method)

Sample 6, 500V, 45min, $\Delta\mathrm{T} = 0.2\mathrm{s}$, binning = 1:

[11]:
```
# Messdaten mit ΔT = 0.2, binning = 1
zerfälle_6_500_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
anzahl_6_500_1 =␣
 ↪[3,50,167,477,1056,1707,2308,2482,2159,1493,916,424,165,69,19,2,2]

# Passende Poissonverteilung:
lambda = 7.1
N = sum(anzahl_6_500_1)
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))

#Plot
poisson1 = bar(zerfälle_6_500_1,anzahl_6_500_1,label=L"\mathrm{Messdaten}")
```

```
bar!(0:25,P,color=:green,label=LaTeXString("\$\\mathrm{Poisson\\ mit\\ }␣
  ↪\\lambda = $lambda\$"),alpha=0.5)
xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }0.2\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[11]:



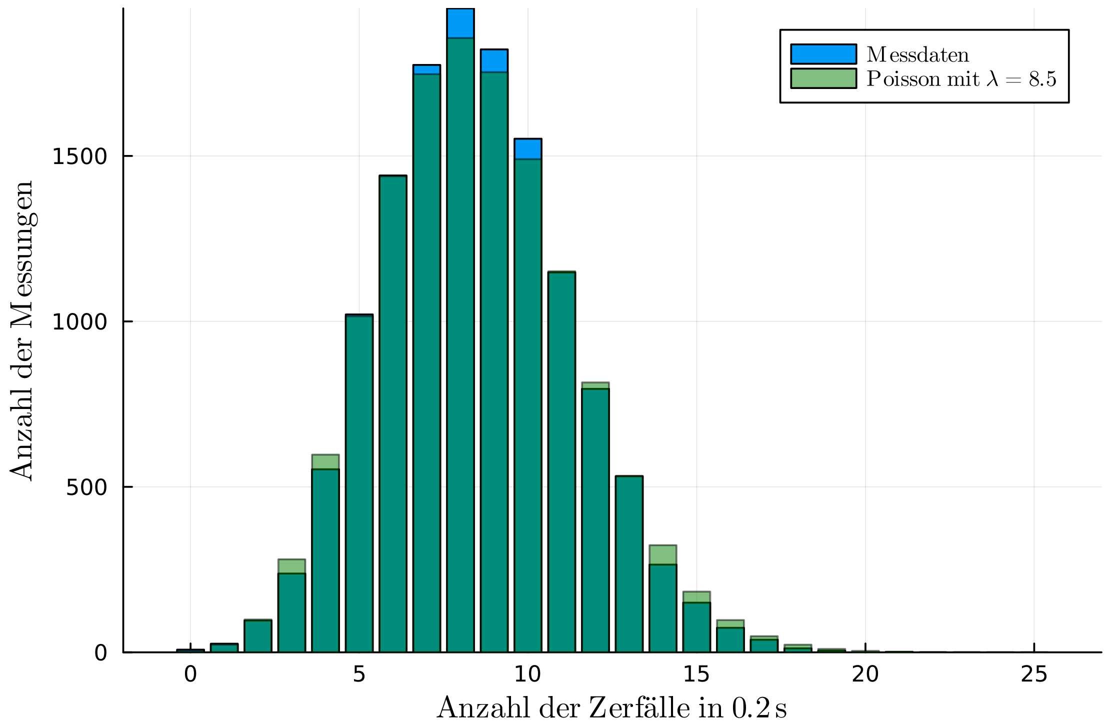
[12]:
```
savefig(poisson1, "../../media/B3.1/poisson1.pdf");
```

Sample 6, 600V, 45min, ΔT = 0.2s, binning = 1:

[13]:
```
# Messdaten mit ΔT = 0.2s, binning = 1
zerfälle_6_600_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]
anzahl_6_600_1 =␣
  ↪[8,26,96,238,553,1021,1441,1775,1946,1822,1552,1148,796,532,265,150,74,38,12,5,0,1]

# Passende Poissonverteilung:
lambda = 8.5
N = sum(anzahl_6_600_1)
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))

# Plot
poisson2 = bar(zerfälle_6_600_1,anzahl_6_600_1,label=L"\mathrm{Messdaten}")
bar!(0:25,P,color=:green,label=LaTeXString("\$\\mathrm{Poisson\\ mit\\ }␣
  ↪\\lambda = $lambda\$"),alpha=0.5)
```

```
xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }0.2\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[13]:



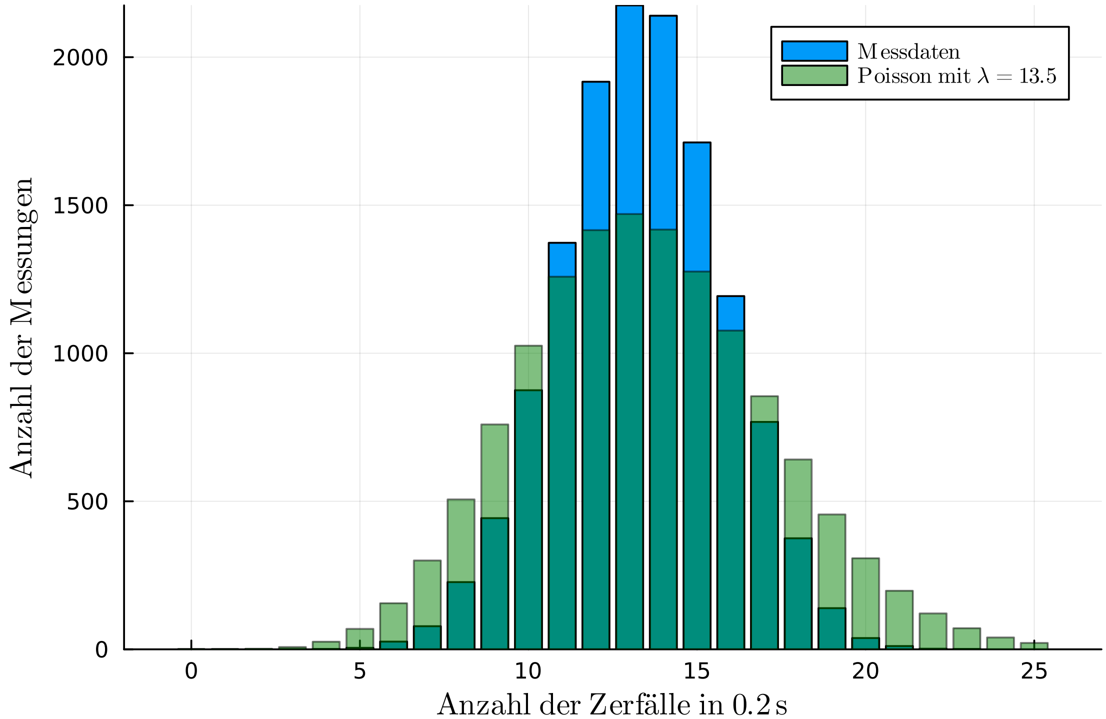
[14]:
```
savefig(poisson2, "../../media/B3.1/poisson2.pdf");
```

Sample 6 + 7, 500V, 45min, $\Delta T = 0.2s$, binning = 1:

[15]:
```
# Messdaten mit ΔT = 0.2s, binning = 1
zerfälle_6und7_500_1 =
  [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]
anzahl_6und7_500_1 =
  [0,0,0,0,1,5,26,78,227,443,875,1373,1917,2175,2140,1712,1193,768,375,139,38,11,2,1]

# Passende Poissonverteilung:
lambda = 13.5
N = sum(anzahl_6und7_500_1)
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))

# Plot
poisson3 =
  bar(zerfälle_6und7_500_1,anzahl_6und7_500_1,label=L"\mathrm{Messdaten}")
bar!(0:25,P,color=:green,label=LaTeXString("\$\\mathrm{Poisson\\ mit\\ }
  \\lambda = $lambda\$"),alpha=0.5)
```

```
xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }0.2\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[15]:



[16]:

```
savefig(poisson3, "../../media/B3.1/poisson3.pdf");
```

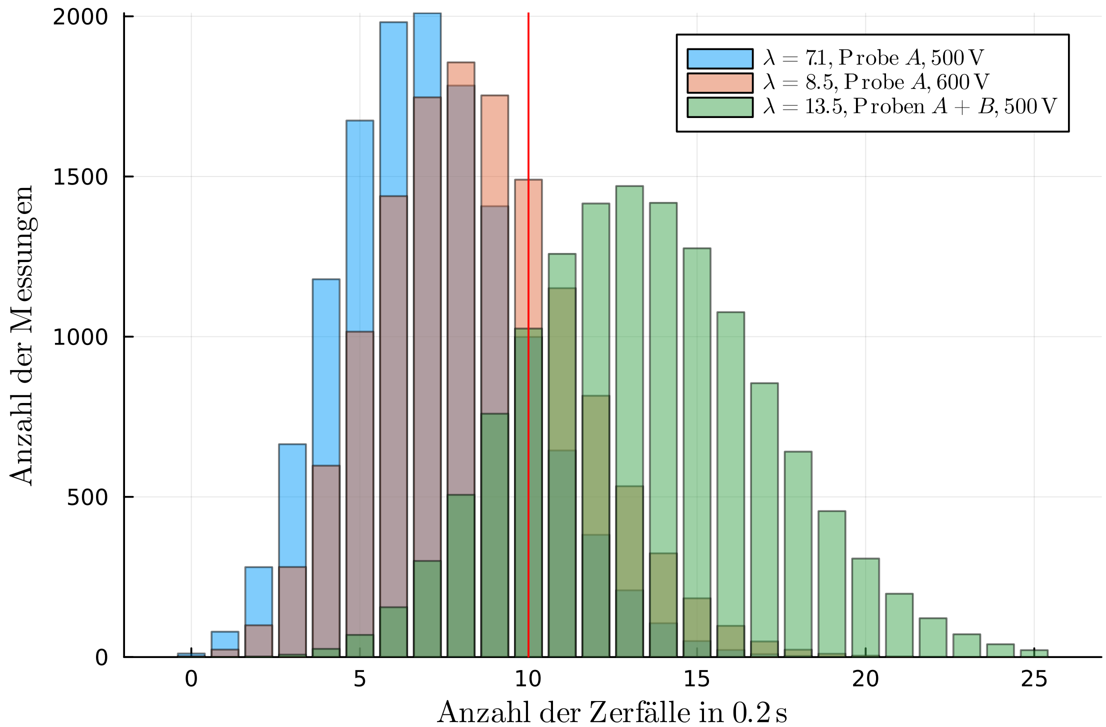
Alle Poisson-Verteilungen zusammen:

[17]:

```
# Messdaten mit ΔT = 0.2, binning = 1
zerfälle_6_500_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
anzahl_6_500_1 =␣
 ↪[3,50,167,477,1056,1707,2308,2482,2159,1493,916,424,165,69,19,2,2]

# Passende Poissonverteilung:
lambda = 7.1
N = sum(anzahl_6_500_1)
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))

#Plot
allePoisson = bar(0:25,P,label=LaTeXString("\$\\lambda = $lambda,␣
 ↪\\mathrm{Probe\\ } A, 500\\mathrm{\\,V}\$"),alpha=0.5)

# Messdaten mit ΔT = 0.2s, binning = 1
zerfälle_6_600_1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]
```

```
anzahl_6_600_1 =
↪[8,26,96,238,553,1021,1441,1775,1946,1822,1552,1148,796,532,265,150,74,38,12,5,0,1]

# Passende Poissonverteilung:
lambda = 8.5
N = sum(anzahl_6_600_1)
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))

# Plot
bar!(0:25,P,label=LaTeXString("\$\\lambda = $lambda, \\mathrm{Probe\\ } A,
↪600\\mathrm{\\,V}\$"),alpha=0.5)

# Messdaten mit ΔT = 0.2s, binning = 1
zerfälle_6und7_500_1 =
↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]
anzahl_6und7_500_1 =
↪[0,0,0,0,1,5,26,78,227,443,875,1373,1917,2175,2140,1712,1193,768,375,139,38,11,2,1]

# Passende Poissonverteilung:
lambda = 13.5
N = sum(anzahl_6und7_500_1)
P(n) = N * lambda^big(n) * exp(-lambda) / factorial(big(n))

# Plot
bar!(0:25,P,label=LaTeXString("\$\\lambda = $lambda, \\mathrm{Proben\\ } A+B,
↪500\\mathrm{\\,V}\$"),alpha=0.5)

xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }0.2\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")

# Plot Faustregel
vline!([10], color=:red, label="")
```

[17]:



[18]:
```
savefig(allePoisson, "../../media/B3.1/allePoisson.pdf");
```

1.4 2: Gaußverteilung

[19]:
```
n = 8 # Binning (für alle Messungen gleich)
Δt = 5 # (für alle Messungen gleich)

G(x, m, F, n=8) = 1/sqrt(2 * pi * m) * F * sqrt(n) * exp(- (x - m)^2 / (2 * m /
  n))
```

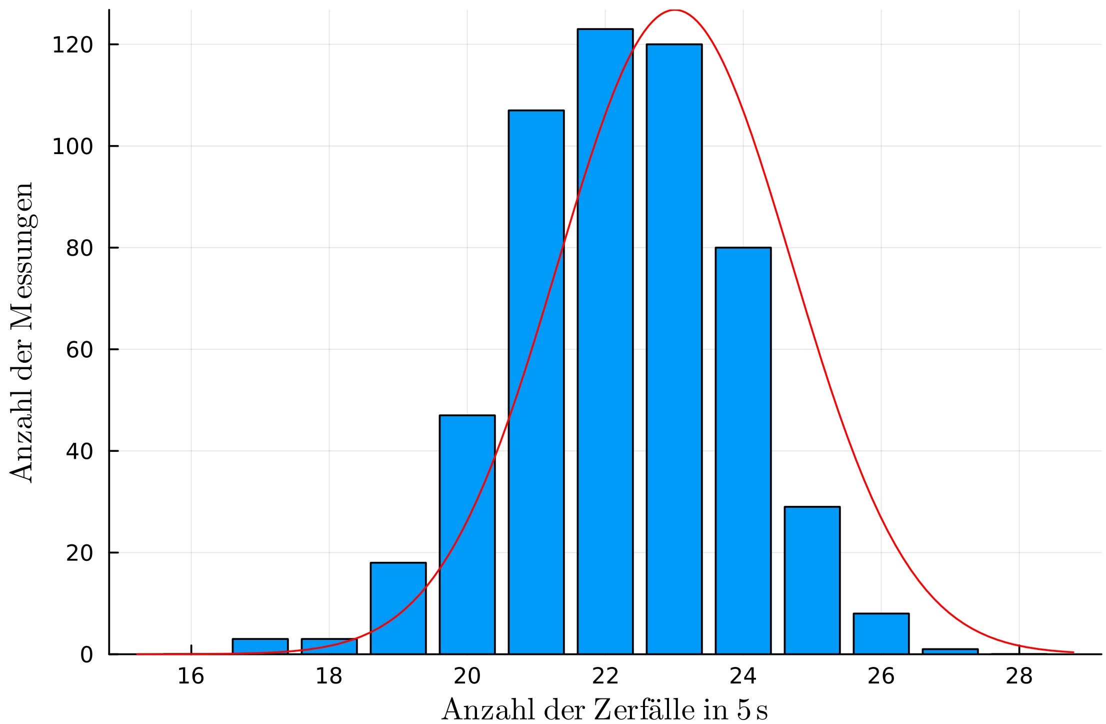
[19]: `G (generic function with 2 methods)`

Sample 6, 500V, 45min, ΔT = 5s, binning = 8:

[20]:
```
# Messdaten mit ΔT = 5s, binning = 8
anzahl_6_500_2 =
  [2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,18,47,107,123,120,80,29,8,1,0,0,0]

# Schneide interessanten Bereich hereaus
zerfälle_6_500_2 = 16:28
anzahl_6_500_2 = anzahl_6_500_2[zerfälle_6_500_2]

# Passende Gaußverteilung:
```

```
z_strich = 11044/300 # Aus Kurzmessung
m = z_strich * Δt / n
F = sum(anzahl_6_500_2)
G(x) = G(x, m, F)

# Plot
gauß1 = plot(bar(zerfälle_6_500_2,anzahl_6_500_2,label=L"\mathrm{Messdaten}"),␣
 ↪legend=:none)
plot!(G,color=:red,label=L"\mathrm{Gauß}")

xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }5\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[20]:



[21]:

```
[m,F,z_strich]
```

[21]:

```
3-element Vector{Float64}:
  23.008333333333333
 539.0
  36.81333333333333
```

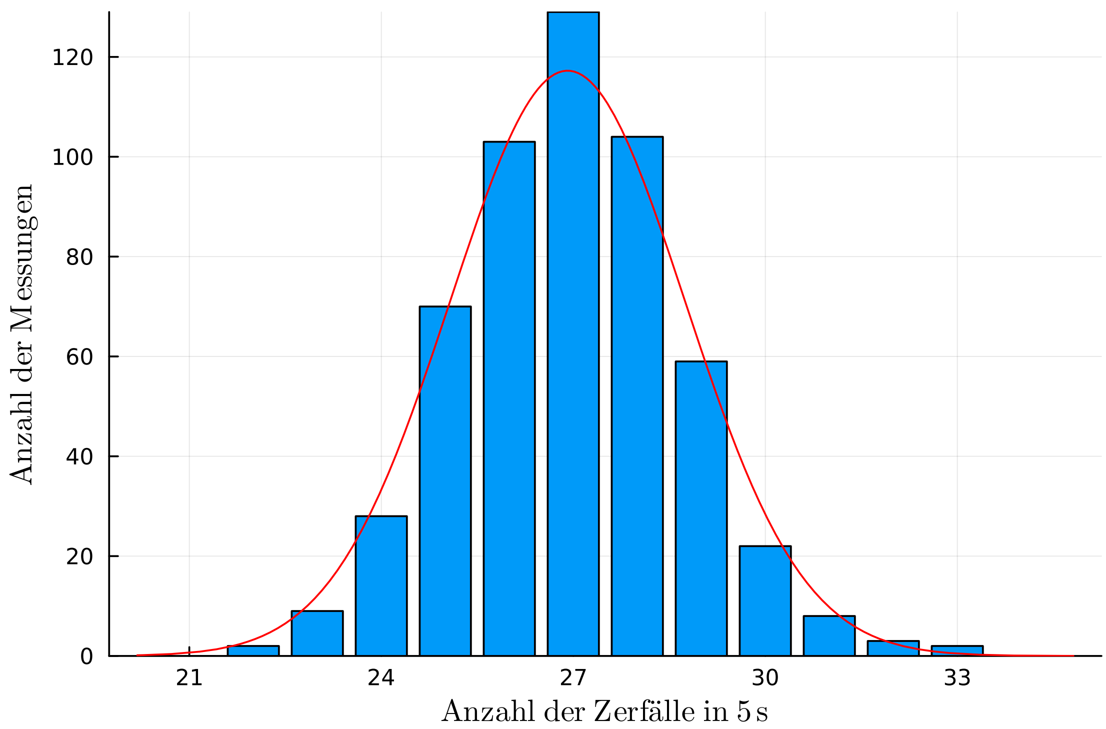
[22]:

```
savefig(gauß1, "../../media/B3.1/gauss1.pdf");
```

Sample 6, 600V, 45min, $\Delta T = 5s$, binning = 8:

[23]:
```julia
# Messdaten mit ΔT = 5s, binning = 8
anzahl_6_600_2 =
 ↪[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,9,28,70,103,129,104,59,22,8,3,2,0,0]

# Schneide interessanten Bereich hereaus
zerfälle_6_600_2 = 21:34
anzahl_6_600_2 = anzahl_6_600_2[zerfälle_6_600_2]

# Passende Gaußverteilung:
z_strich = 12917/300 # Aus Kurzmessung
m = z_strich * Δt / n
F = sum(anzahl_6_600_2)
G(x) = G(x, m, F)

# Plot
gauß2 = plot(bar(zerfälle_6_600_2,anzahl_6_600_2,label=L"\mathrm{Messdaten}"),
 ↪legend=:none)
plot!(G,color=:red,label=L"\mathrm{Gauß}")
xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }5\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[23]:



[24]:
```julia
[m,F,z_strich]
```

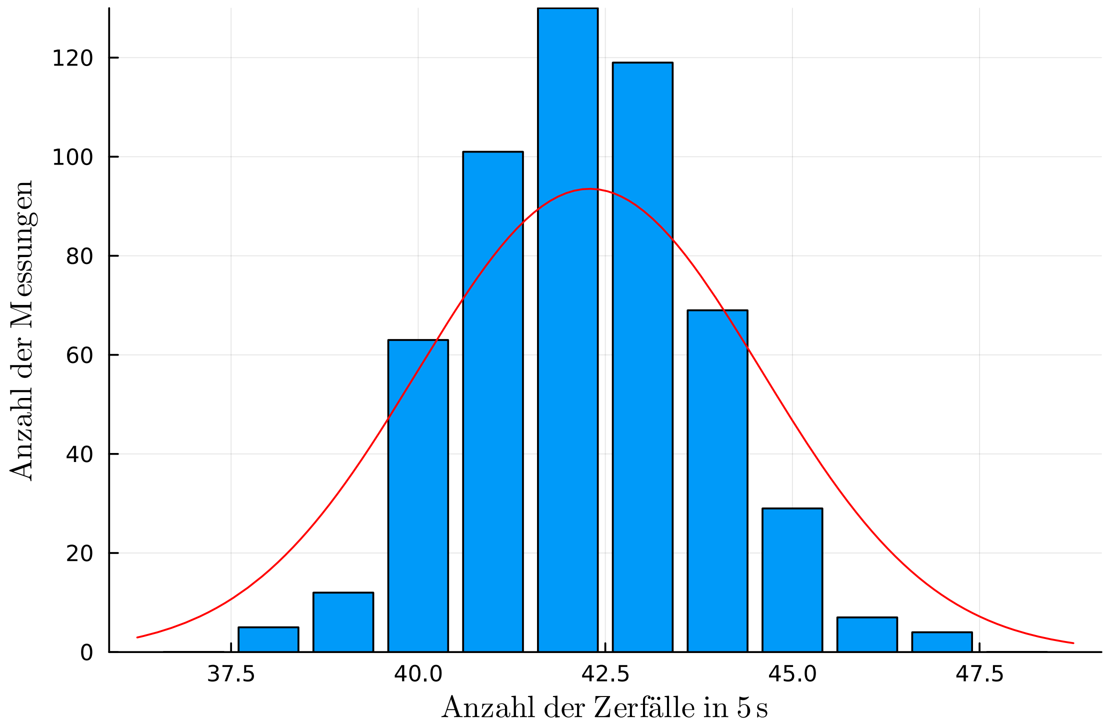
[24]:
```
3-element Vector{Float64}:
  26.910416666666666
 539.0
  43.056666666666665
```

[25]:
```julia
savefig(gauß2, "../../media/B3.1/gauss2.pdf");
```

Sample 6 + 7, 500V, 45min, ΔT = 5s, binning = 8:

[26]:
```julia
# Messdaten mit ΔT = 5s, binning = 8
anzahl_6und7_500_2 =␣
 ↪[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,12,63,101,130,
   29,7,4,0,0]

# Schneide interessanten Bereich hereaus
zerfälle_6und7_500_2 = 37:48
anzahl_6und7_500_2 = anzahl_6und7_500_2[zerfälle_6und7_500_2]



# Passende Gaußverteilung:
z_strich = 20300/300 # Aus Kurzmessung
m = z_strich * Δt / n
F = sum(anzahl_6und7_500_2)
G(x) = G(x, m, F)

# Plot
gauß3 =␣
 ↪plot(bar(zerfälle_6und7_500_2,anzahl_6und7_500_2,label=L"\mathrm{Messdaten}"),␣
 ↪legend=:none)
plot!(G,color=:red,label=L"\mathrm{Gauß}")
xlabel!(L"\mathrm{Anzahl\ der\ Zerfälle\ in\ }5\mathrm{\,s}")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[26]:



[27]:
```
[m,F,z_strich]
```

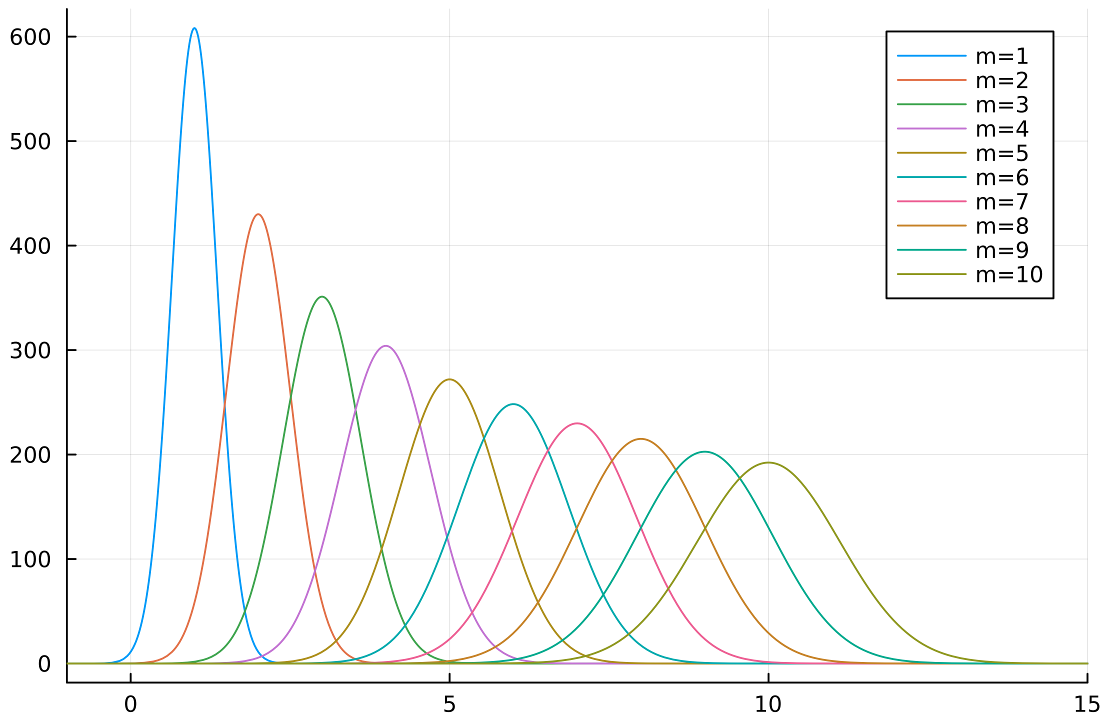
[27]:
```
3-element Vector{Float64}:
  42.29166666666667
 539.0
  67.66666666666667
```

[28]:
```
savefig(gauß3, "../../media/B3.1/gauss3.pdf");
```

[29]:
```
G(x) = G(x, m, F)
m = 1
plot(G, xaxis=[-1,15], label="m=$m")
m=2
plot!(G, label="m=$m")
m=3
plot!(G, label="m=$m")
m=4
plot!(G, label="m=$m")
m=5
plot!(G, label="m=$m")
m=6
plot!(G, label="m=$m")
m=7
```

```
plot!(G, label="m=$m")
m=8
plot!(G, label="m=$m")
m=9
plot!(G, label="m=$m")
m=10
plot!(G, label="m=$m")
```

[29]:



1.5 3: Intervall-Verteilung

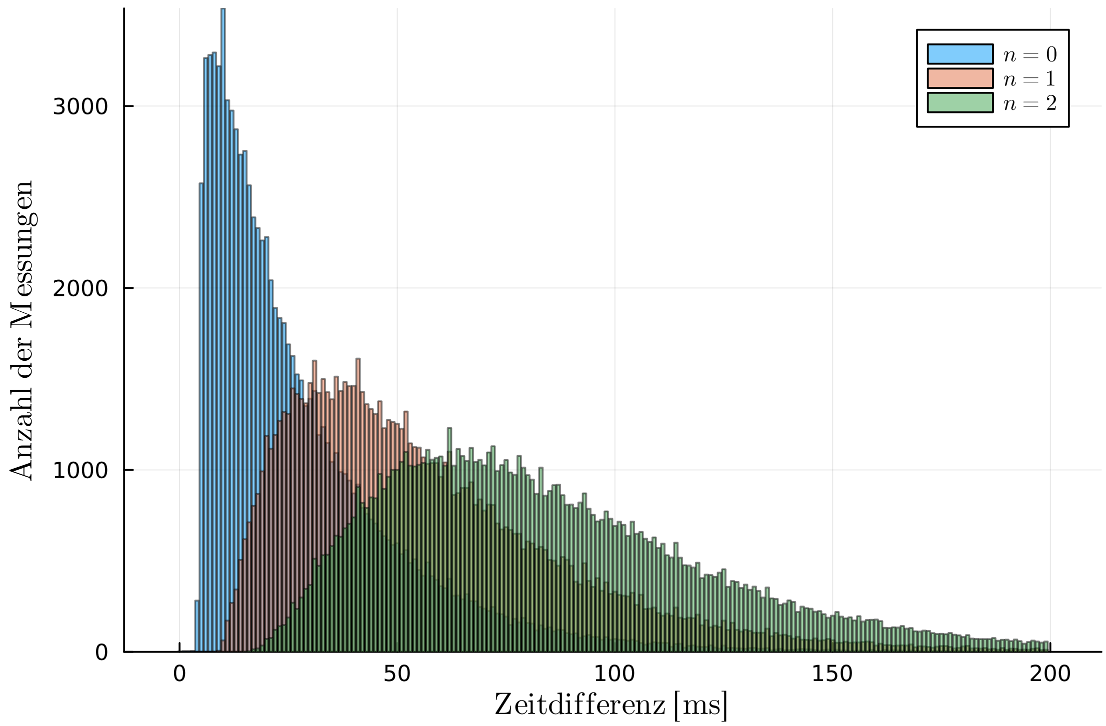
Plotten der Messwerte

[30]:
```
interval0 = CSV.read("sample_6/500V_45min/interval_0.001_1.csv", DataFrame)
interval1 = CSV.read("sample_6/500V_45min/interval_0.001_2.csv", DataFrame)
interval2 = CSV.read("sample_6/500V_45min/interval_0.001_3.csv", DataFrame)
;
```

[31]:
```
# Sample 6, 500V, 45min
interval = bar(interval0[1:200, :intervall], interval0[1:200, :count], alpha=0.
 ↪5, label=L"n = 0")
bar!(interval1[1:200, :intervall], interval1[1:200, :count], alpha=0.5,␣
 ↪label=L"n = 1")
bar!(interval2[1:200, :intervall], interval2[1:200, :count], alpha=0.5,␣
 ↪label=L"n = 2")
```

```
xlabel!(L"\mathrm{Zeitdifferenz\ } [\mathrm{ms}]")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[31]:



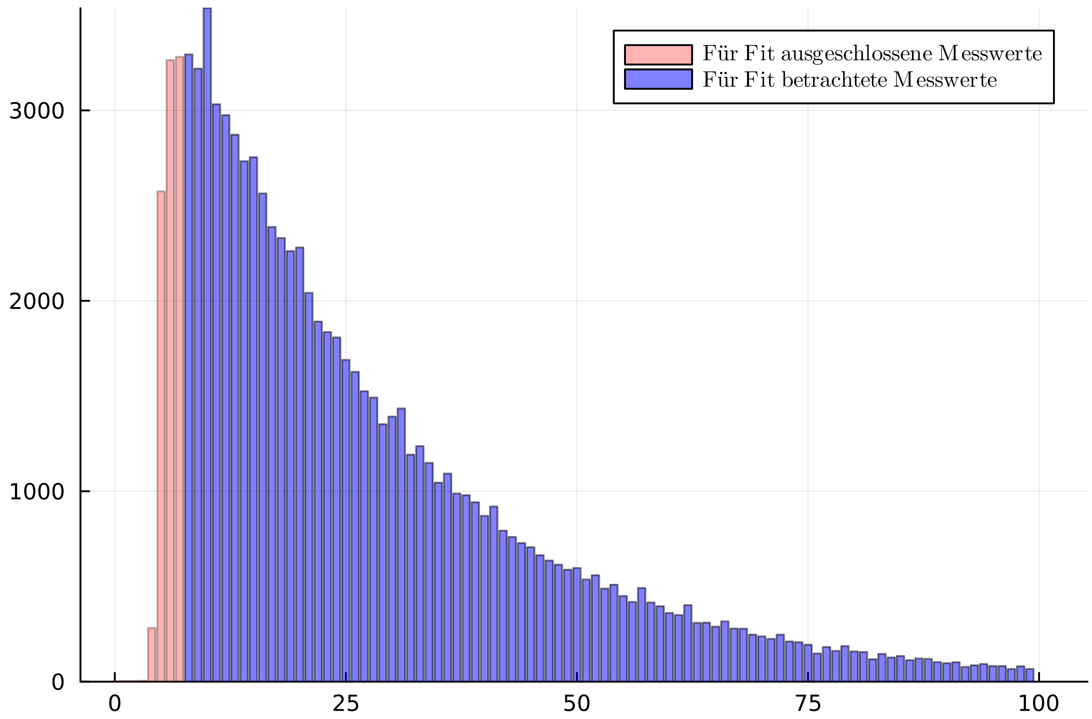
[32]:
```
savefig(interval, "../../media/B3.1/interval.pdf");
```

Fitten für n=0

- Alle Messwerte im Bereich t ∈ [0,totzeit] abschneiden und den Rest fitten

[33]:
```
# Plot
totzeitIndex = 8
intervalFit = bar(interval0[1:totzeitIndex, :intervall], interval0[1:
 ↪totzeitIndex, :count],
    color=:red, alpha=0.3, label=L"\mathrm{Für\ Fit\ ausgeschlossene\␣
 ↪Messwerte}", title="", legend=:topright)
bar!(interval0[(totzeitIndex+1):100, :intervall],
    interval0[(totzeitIndex+1):100, :count], color=:blue, alpha=0.5,␣
 ↪label=L"\mathrm{Für\ Fit\ betrachtete\ Messwerte}", title="")
```

[33]:

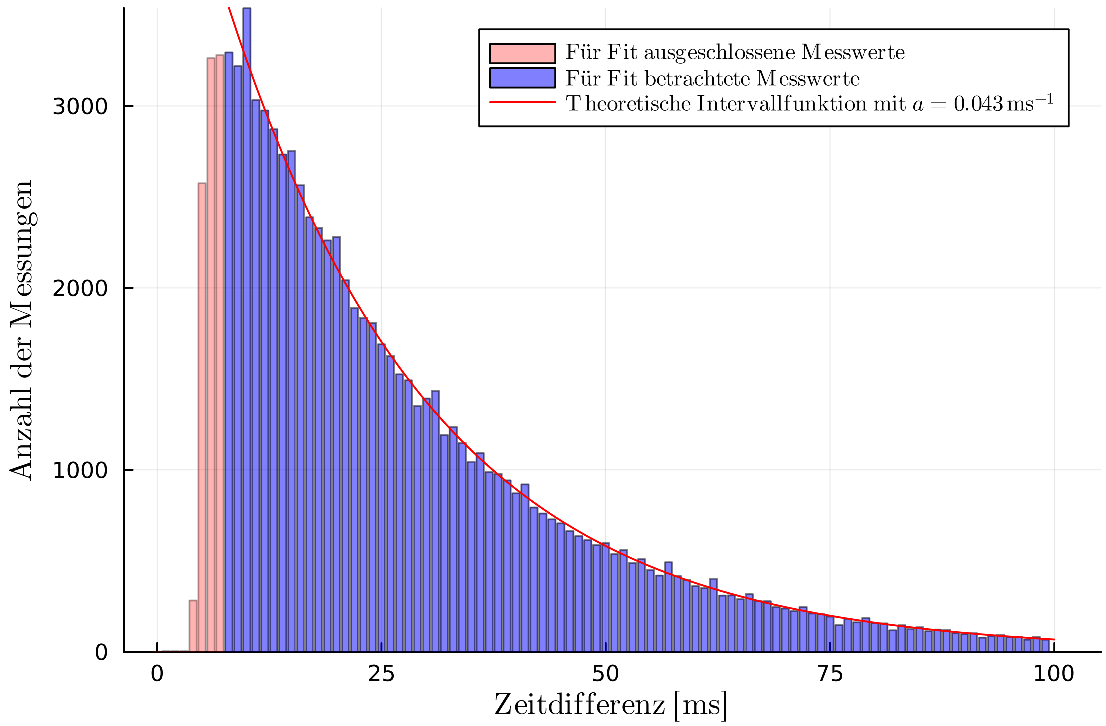


[34]:
```
# Intervallverteilung mit skaliertem N (für n = 0)
a = 0.043 # Fitparameter zwischen 0.042 und 0.044
T = 2700 # s Gesamtdauer der Messung = 45 min oder letzter gemessener Zeitpunkt?
N = 1000*T * a # Normierungskonstante
I(t) = N * a * exp(- a * t)
```

[34]: I (generic function with 1 method)

[35]:
```
plot!(totzeitIndex:100,I,label=LaTeXString("\$\\mathrm{Theoretische\\␣
↪Intervallfunktion\\ mit\\ } a = $a \\mathrm{\\,ms^{-1}}\$"), color=:red)

xlabel!(L"\mathrm{Zeitdifferenz\ } [\mathrm{ms}]")
ylabel!(L"\mathrm{Anzahl\ der\ Messungen}")
```

[35]:



[36]:
```
T/1000
```

[36]: 2.7

[37]:
```
savefig(intervalFit, "../../media/B3.1/intervalFit.pdf");
```

1.6 4: Totzeit

$$a = a'/(1 - a' * \tau)$$
$$\Leftrightarrow \tau = 1/a' - 1/a$$

Benötigt a aus der Intervallverteilung. Falls a nicht schon definiert ist, wird es hier definiert:

[2]:
```
if ! isdefined(Main, :a)
    a = 0.043
end
```

[2]: 0.043

[3]:
```
a_strich = 11044/300 * 10^(-3) # ms Gemessene Zählrate aus Kurzzeitmessung
tau = 1/a_strich - 1/a # ms
```

[3]: 3.908257035283807

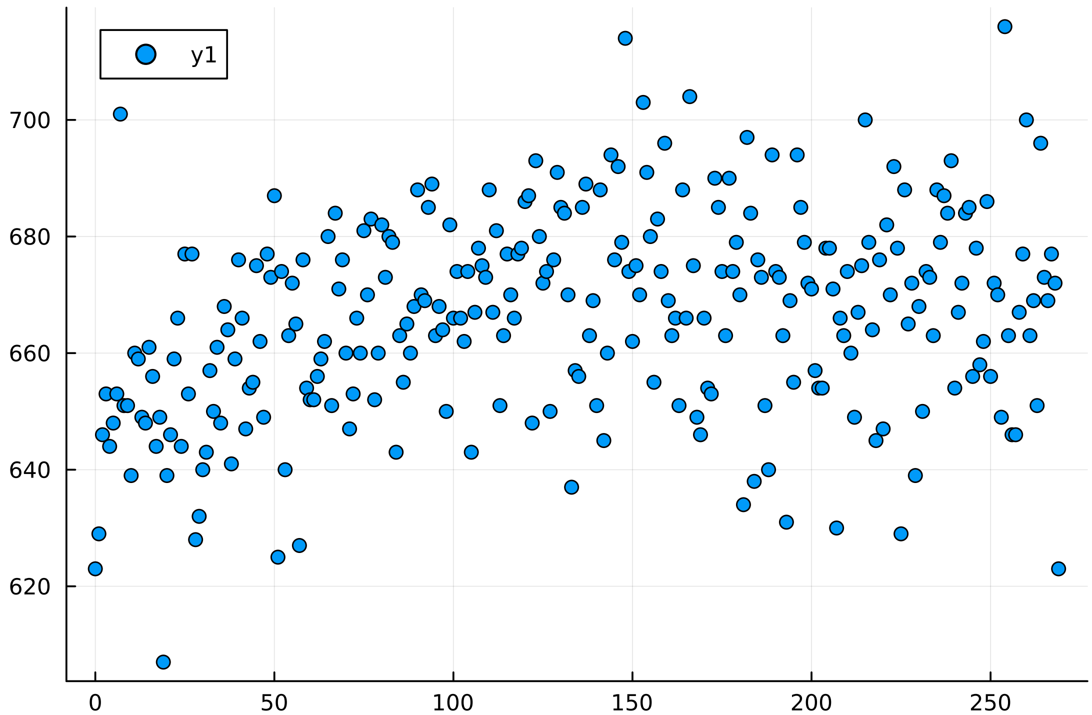
1.7 5: Aufgaben zum χ^2–Test

Benötigt Totzeit.

Daten einlesen

[4]:
```
data = CSV.File("sample_6_7/500V_45min/divide_10.csv")
scatter(0:length(data), data.Zerfälle_in_10_sec)
```

[4]:



[5]:
```
pos_min = 0
pos_max = 50
x_data = pos_min:pos_max
decays = data.Zerfälle_in_10_sec[x_data.+1]

avg_decays = mean(decays)
```

[5]: 653.8039215686274

[6]:
```
std(decays)
```

[6]: 16.509415020336895

[7]:
```
avg_decays
```

[7]: 653.8039215686274

[8]:
```
avg_decays_corrected = avg_decays / (1 - a_strich*tau)
```

[8]:
```
763.6789739437968
```

[9]:
```
avg_decays_corrected/avg_decays
```

[9]:
```
1.1680550525172038
```

[10]:
```
decays_corrected = decays ./ (1 .- a_strich*tau);
```

1.7.1 1. Hypothesen

Zeigen Sie zeichnerisch, was die Hypothesen a, b und c besagen.

- a: Die Präparatstärke ist konstant im betrachteten Zeitraum und gleicht dem Mittelwert der 51 Messwerte.
- b: Die Präparatstärke ist konstant im betrachteten Zeitraum und gleicht dem Mittelwert der 51 Messwerte minus 10%.
- c: Die Präparatstärke nimmt im betrachteten Zeitraum linear mit der Zeit ab (als erste Näherung eines exponentiellen Abfalls). Die Anfangszählrate ist der Mittelwert, und der Abfall von einer Messung zur anderen sei 1.

Hypothese H_1: "Die Präparatstärke ist konstant im betrachteten Zeitraum und gleicht dem Mittelwert der 51 Messwerte."

Diese Hypothese besagt, jeder erwartete Zählmenge $n_1(i)$ sei gleich dem Mittelwert der 51 Messungen n_i.

$$\bar{x} = \bar{n}$$

$$n_1(i) = \frac{1}{51}\sum_{i=1}^{51} n_i$$

$$\chi_1^2 = \sum_i \frac{(n_i - \bar{n})^2}{\bar{n}}$$

Hypothese H_2: "Die Präparatstärke ist konstant im betrachteten Zeitraum und gleicht dem Mittelwert der 51 Messwerte minus 10%."

Dies bedeutet, dass die erwarteten Zählungen $n_2(i)$ um 10% kleiner als die Zählungen nach H_1 sein müssen.

$$n_2(i) = \frac{9}{10} \cdot n_1(i)$$

$$\chi_2^2 = \sum_i \frac{(n_i - 0.9\,\bar{n})^2}{0.9\,\bar{n}}$$

Hypothese H_3: "Die Präparatstärke nimmt im betrachteten Zeitraum linear mit der Zeit ab (als erste Näherung eines exponentiellen Abfalls). Die Anfangszählrate ist der Mittelwert, und der Abfall von einer Messung zur anderen sei 1."

Die erwarteten Ereignisse starten demnach bei $n_1(0)$ und fallen dann linear mit einer Steigung von 1 ab.

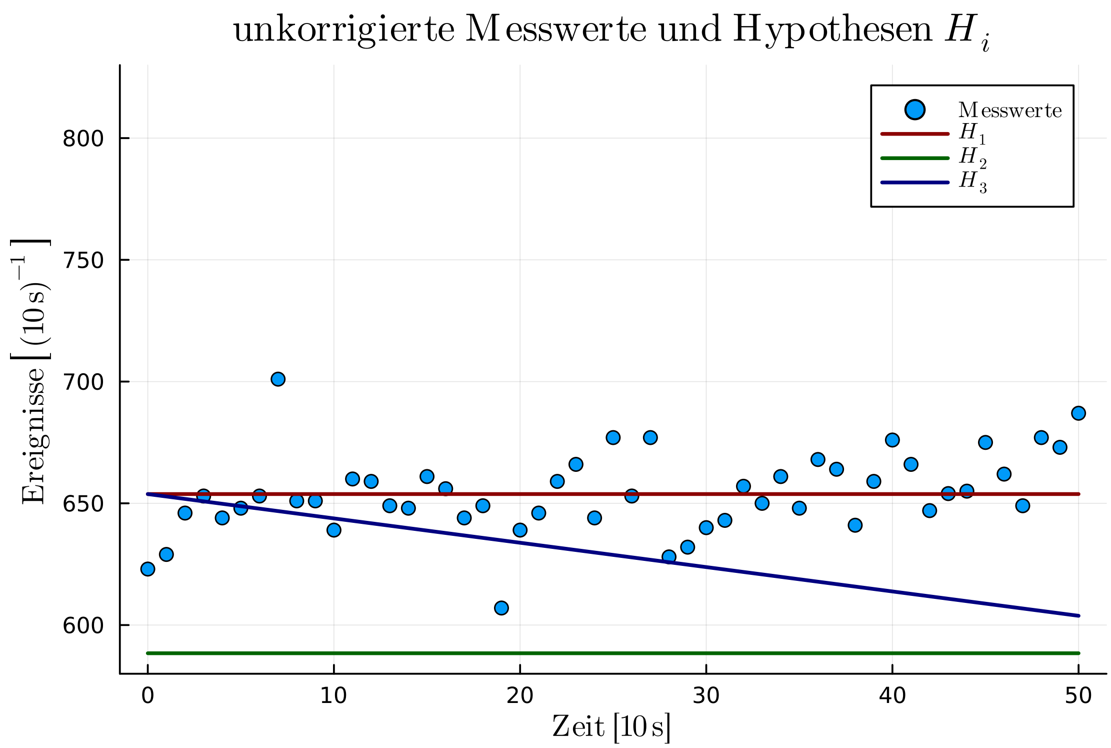
$$n_3(i) = n_1(0) - i$$
$$\chi_3^2 = \sum_i \frac{(n_i - (n-i))^2}{(n-i)}$$

[11]:
```
function plot_hypotheses(x_data, decays, title; legend_pos)
    scatter(x_data, decays, label=L"\mathrm{Messwerte}", legend=legend_pos,␣
 ↪yrange=[580, 830])

    title!(title)
    xlabel!(L"\mathrm{Zeit}\ [10\mathrm{\,s}]")
    ylabel!(L"\mathrm{Ereignisse}\ \left[(10\,\mathrm{s})^{-1}\right]")

    h1_results = fill(mean(decays), length(x_data))

    plot!(
        x_data,
        h1_results,
        label=L"H_1",
        color=:red4,
        lw=2
    )

    plot!(
        x_data,
        0.9 .* h1_results,
        label=L"H_2",
        color=:darkgreen,
        lw=2
    )

    plot!(
        x_data,
        h1_results .- x_data,
        label=L"H_3",
        color=:navy,
        lw=2
    )
end
```

[11]:
```
plot_hypotheses (generic function with 1 method)
```

[12]:
```
plot_hypotheses(x_data, decays, L"\mathrm{unkorrigierte\ Messwerte\ und\␣
 ↪Hypothesen\ } H_i", legend_pos=:topright)
```

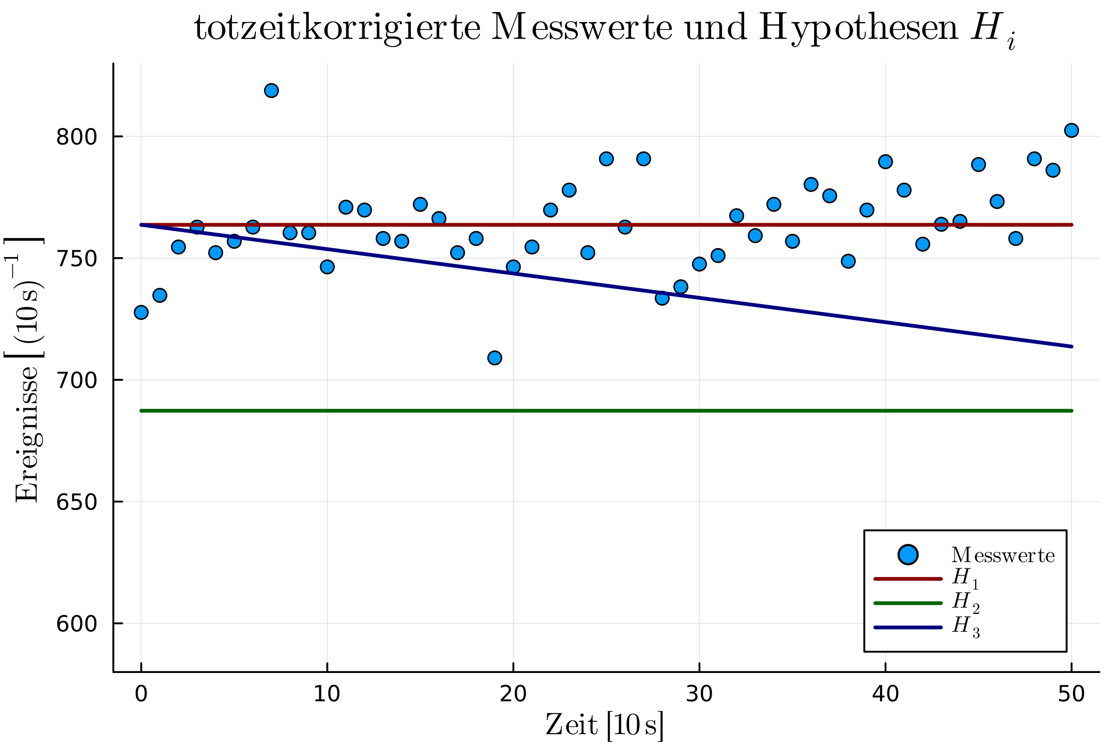
[12]:



[15]:
```
savefig("../../media/B3.1/Hypothesen_plot.svg")
savefig("../../media/B3.1/Hypothesen_plot.pdf");
```

[13]:
```
plot_hypotheses(x_data, decays_corrected, L"\mathrm{totzeitkorrigierte\␣
↪Messwerte\ und\ Hypothesen\ } H_i"; legend_pos=:bottomright)
```

[13]:




```
[17]: savefig("../../media/B3.1/Hypothesen_plot_corr.svg")
      savefig("../../media/B3.1/Hypothesen_plot_corr.pdf");
```

Man Erkennt, dass H_1 die Messwerte am besten beschreibt.

1.7.2 2. Hypothesentest

Nun werden die drei Hypothesen H_i mithilfe des χ^2–Tests geprüft. Hierzu werden 51 Messwerte aus der 45 min-Messung beider Proben gewählt, deren Zählungen über 10 s gemittelt werden. Dies wird sowohl für die nicht-totzeitkorrigierten als auch für die totzeitkorrigierten Daten durchgeführt.

Durch die Bildung des Mittelwertes gibt es noch 50 statistische Freiheitsgrade. Dadurch können die erlaubten Grenzen für χ^2 für ein System mit 50 Freiheitsgraden und einer Signifikanz von 5% verwendet werden [6].

$$\chi^2_{\text{min}} = 32.357$$
$$\chi^2_{\text{max}} = 71.420$$

unkorrigiert

$$\chi^2_1 = \sum_i \frac{(x_i - \bar{x})^2}{\bar{x}}$$

$$\chi^2_2 = \sum_i \frac{(x_i - 0.9\,\bar{x})^2}{0.9\,\bar{x}}$$

$$\chi_3^2 = \sum_i \frac{(x_i - (\bar{n} - i))^2}{(\bar{n} - i)}$$

[14]:
```
chi_squared_1 = sum((decays .- avg_decays).^2 ./ avg_decays)
```

[14]: 20.84422984644914

[15]:
```
(1 - chi_squared_1 / 32.357)*100
```

[15]: 35.58046219844504

[16]:
```
chi_squared_2 = sum((decays .- 0.9*avg_decays).^2 ./ (0.9*avg_decays))
```

[16]: 393.6491442738329

[17]:
```
chi_squared_2 / 71.420
```

[17]: 5.511749429765232

[18]:
```
chi_squared_3 = sum((decays .- (avg_decays .- x_data)).^2 ./ (avg_decays .-
↪x_data))
```

[18]: 106.02106295610936

[19]:
```
chi_squared_3 / 71.420
```

[19]: 1.4844730181477088

totzeitkorrigiert

$$\chi_{i,\mathrm{korr}}^2 = \frac{1}{1 - \frac{m}{\Delta t}\tau} \cdot \chi_i^2$$

[22]:
```
chi_squared_1_corrected = 1 / (1-a_strich*tau) * chi_squared_1
```

[22]: 24.34720798797481

[23]:
```
(1 - chi_squared_1_corrected / 32.357)*100
```

[23]: 24.75443339007074

[24]:
```
chi_squared_2_corrected = 1 / (1-a_strich*tau) * chi_squared_2
```

[24]: 459.80387188812415

[25]:
```
chi_squared_2_corrected / 71.420
```

[25]: 6.438026769646095

[27]:
```
chi_squared_3_corrected = 1 / (1-a_strich*tau) * chi_squared_3
```

[27]: 123.83843825912807

[28]:
```
chi_squared_3_corrected / 71.420
```

[28]: 1.7339462091728937

1.7.3 3. Halbwertszeit

Welche Halbwertszeit ergibt sich aus der Hypothese c, wenn Sie einen zeitlichen Abstand der Messungen von 10 Sekunden annehmen als Näherung eines exponentiellen Zerfalls?

Nun soll nach Hypothese H_3 die Halbwertszeit $T_{1/2}$ der Proben bestimmt werden. Hierzu wird der exponentielle Zerfall durch eine lineare Kurve beschrieben. Es wird erwartet, dass diese Halbwertszeit deutlich geringer als die ca. 30 yr der Probe ist.

Dabei wird davon ausgegangen, dass innerhalb der Messdauer $\Delta t = 10\,\mathrm{s}$ der Wert N_0 um 1 sinkt. Weiterhin wird die Halbwertszeit nach Gleichung **??** bestimmt. Nach Hypothese H_3 ist $N(0) = \bar{n}$ der Mittelwert der Zählraten.

[29]:
```
T_12 = Integer(round(10 * log(2) / log(avg_decays/(avg_decays-1)))) # seconds
println(T_12, " seconds")

T_12_str = string("\$T_{1/2}(m^\\prime)=", Integer(round(T_12/3600)),␣
↪"\\,\\mathrm{h}\\,", Integer(round(T_12%60)), "\\,\\mathrm{min}\$",)
println(T_12_str)
latexstring(T_12_str)
```

```
4528 seconds
$T_{1/2}(m^\prime)=1\,\mathrm{h}\,28\,\mathrm{min}$
```

[29]: $T_{1/2}(m') = 1\,\mathrm{h}\,28\,\mathrm{min}$

[30]:
```
T_12_corr = Integer(round(10 * log(2) / log(avg_decays_corrected/
↪(avg_decays_corrected-1)))) # seconds
println(T_12_corr, " seconds")

T_12_corr_str = string("\$T_{1/2}(m)=", Integer(round(T_12_corr/3600)),␣
↪"\\,\\mathrm{h}\\,", Integer(round(T_12_corr%60)), "\\,\\mathrm{min}\$",)
println(T_12_corr_str)
latexstring(T_12_corr_str)
```

```
5290 seconds
$T_{1/2}(m)=1\,\mathrm{h}\,10\,\mathrm{min}$
```

[30]: $T_{1/2}(m) = 1\,\mathrm{h}\,10\,\mathrm{min}$