Statistik und Qualität - Ausarbeitung der vierten Übung

Stefan Dünser

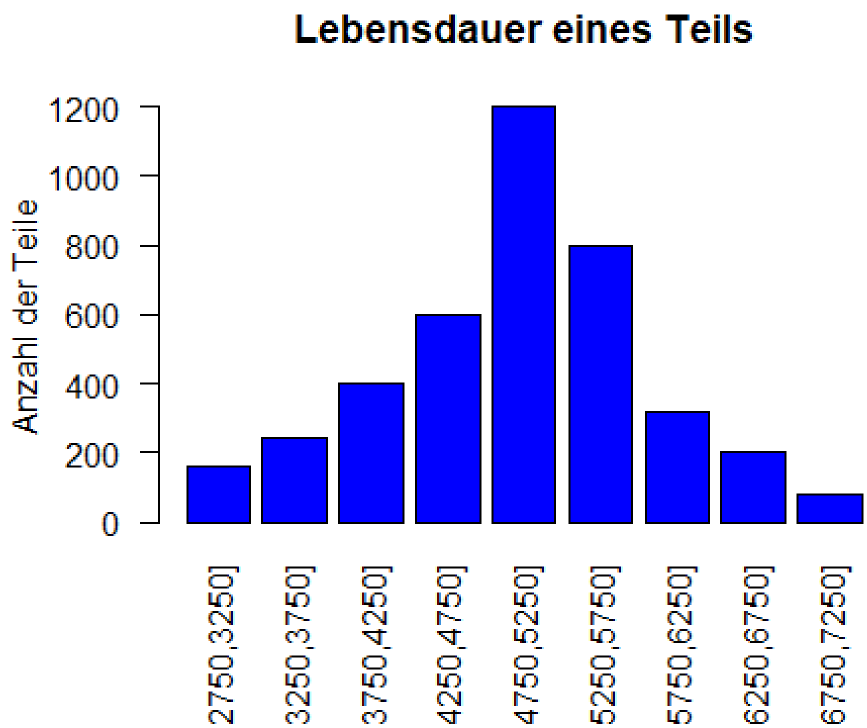
B-1) Ein Produktionsbetrieb arbeitet mit Maschinen, die einen Verschleißteil T enthalten. Die typischen, für den Teil T beobachteten Lebensdauern waren in 4000 Fällen:

```
# Erstellen einer Liste mit den Werten aus der Angabe
Verschleissteile.df <- data.frame(KlassenLebensdauer = c("[2750,3250]","]3250
,3750]","]3750,4250]","]4250,4750]","]4750,5250]","]5250,5750]","]5750,6250]"
,"]6250,6750]", "]6750,7250]"), Klassenbreite = rep(500.00,9), Klassenmitte =
c(3000.00,3500.00,4000.00,4500.00,5000.00,5500.00,6000.00,6500.00,7000.00), A
nzahl = c(160,240,400,600,1200,800,320,200,80))
Verschleissteile.df

##   KlassenLebensdauer Klassenbreite Klassenmitte Anzahl
## 1        [2750,3250]           500         3000    160
## 2        ]3250,3750]           500         3500    240
## 3        ]3750,4250]           500         4000    400
## 4        ]4250,4750]           500         4500    600
## 5        ]4750,5250]           500         5000   1200
## 6        ]5250,5750]           500         5500    800
## 7        ]5750,6250]           500         6000    320
## 8        ]6250,6750]           500         6500    200
## 9        ]6750,7250]           500         7000     80
```

Schätze die Wahrscheinlichkeiten für die Lebensdauer des Teiles T in den gegebenen Klassenbreiten.

Aus der tabellarischen Darstellung geht hervor, dass die meisten Teile in der Mitte der Verteilung zu finden ist (KlasseLebensdauer “]4750,5250]”). Dies deutet darauf hin, dass der Schwerpunkt der Verteilung bei der erwähnten KlassenLebensdauer liegt. Der Schwerpunkt wiederum symbolisiert wiederum das arithmetische Mittel der Vereteilung. Geschätzt liegt das arithmetische Mittel bei rund 5000 h.

```
# Anordnen der Beschriftung der Klassenbreiten (Intervalle) vertikal
par(las = 2)

barplot(Verschleissteile.df$Anzahl, names.arg = Verschleissteile.df$KlassenLe
bensdauer, ylab = "Anzahl der Teile", main = "Lebensdauer eines Teils", col =
"blue")
```



Auch anhand der grafischen Darstellung kann der Schwerpunkt und somit das arithmetische Mittel der Verteilung auf ca. 5000 h geschätzt werden.

Berechne die mittlere Lebensdauer (Erwartungswert) und die Streuung (Standardabweichung) der Lebensdauer (in Stunden). (μ = 4950 Std, s = 867 Std)

Erwartungswert:

```
Erwartungswert <- sum(Verschleissteile.df$Klassenmitte*(Verschleissteile.df$A
nzahl/sum(Verschleissteile.df$Anzahl)))
Erwartungswert

## [1] 4950
```

Standardabweichung:

```
n.vec <- Verschleissteile.df$Anzahl/sum(Verschleissteile.df$Anzahl)
Standardabweichung <- sqrt(sum((Verschleissteile.df$Klassenmitte-Erwartungswe
rt)^2*n.vec))
Standardabweichung

## [1] 867.4676
```

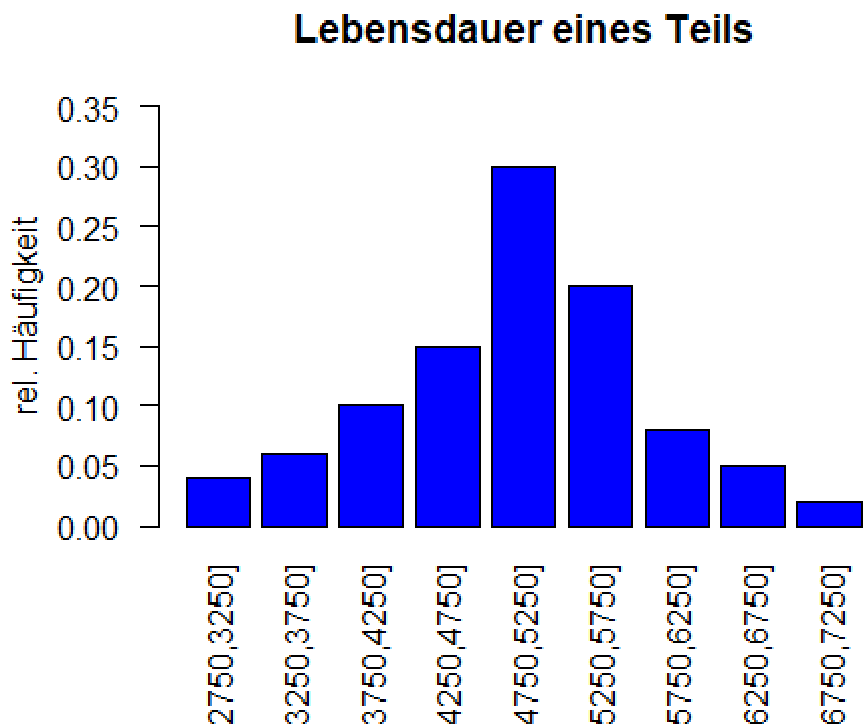
Stelle die Wahrscheinlichkeitsfunktion und die Verteilungsfunktion (Summenhäufigkeit) graphisch dar.

```
# Berechnen der Summenhäufigkeit und der relativen Häufigkeit
SummenHfk.vec <- cumsum(Verschleissteile.df$Anzahl)/sum(Verschleissteile.df$A
nzahl)
Haeufigkeit.vec <- Verschleissteile.df$Anzahl/sum(Verschleissteile.df$Anzahl)
# Ergänzen der Werteliste aus der Angabe um die berechneten Werte
Verschleissteile.df <- data.frame(Verschleissteile.df, SummenHfk = SummenHfk.
vec, Haeufigkeit = Haeufigkeit.vec)
Verschleissteile.df
```

```
##   KlassenLebensdauer Klassenbreite Klassenmitte Anzahl SummenHfk Haeufigke
it
## 1        [2750,3250]           500         3000    160      0.04         0.
04
## 2        ]3250,3750]           500         3500    240      0.10         0.
06
## 3        ]3750,4250]           500         4000    400      0.20         0.
10
## 4        ]4250,4750]           500         4500    600      0.35         0.
15
## 5        ]4750,5250]           500         5000   1200      0.65         0.
30
## 6        ]5250,5750]           500         5500    800      0.85         0.
20
## 7        ]5750,6250]           500         6000    320      0.93         0.
08
## 8        ]6250,6750]           500         6500    200      0.98         0.
05
## 9        ]6750,7250]           500         7000     80      1.00         0.
02
```

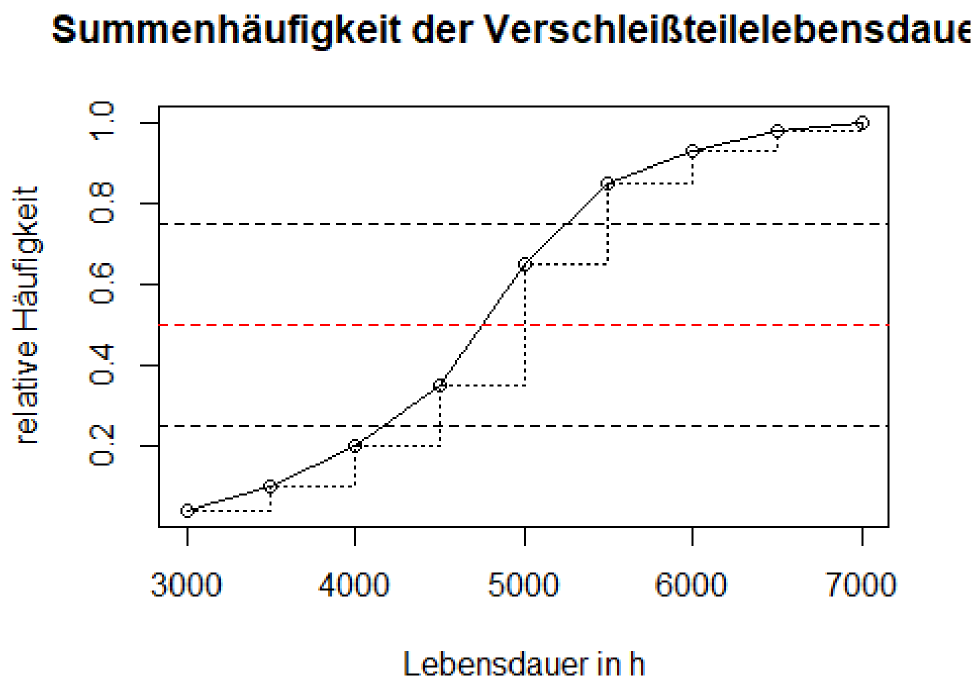
Wahrscheinlichkeitsfunktion:

```
# Anordnen der Beschriftung der Klassenbreiten (Intervalle) vertikal
par(las = 2)

barplot(Verschleissteile.df$Haeufigkeit, names.arg = Verschleissteile.df$Klas
senLebensdauer, ylab = "rel. Häufigkeit", main = "Lebensdauer eines Teils", c
ol = "blue", ylim = c(0.00,0.35))
```



Verteilungsfunktion (Summenhäufigkeit):

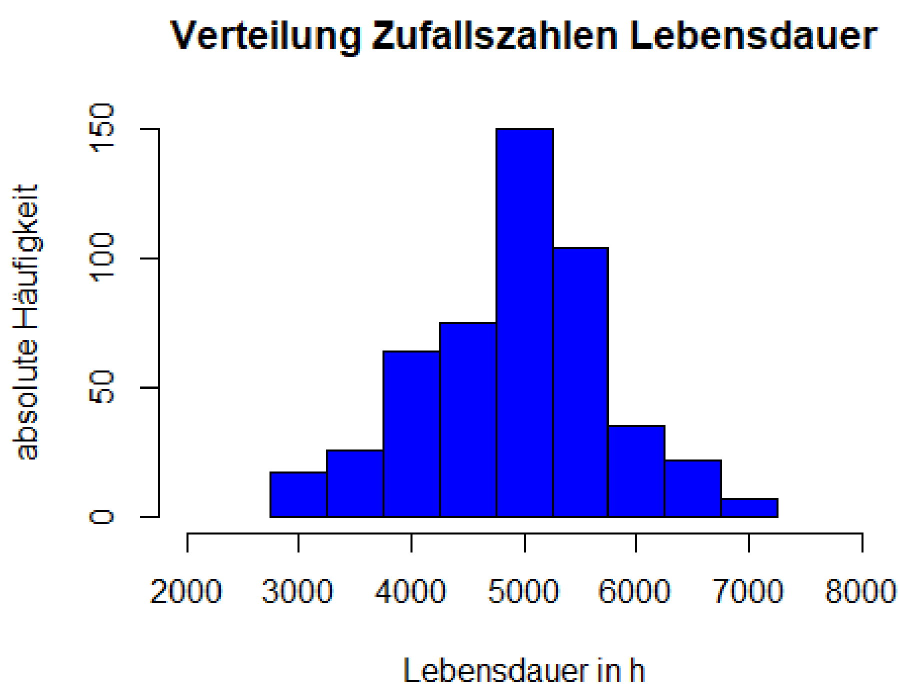
```
Breaks.vec <- c(3000,3500,4000,4500,5000,5500,6000,6500,7000)
plot(Breaks.vec, Verschleissteile.df$SummenHfk,type = "l", lty = 1, , main = 
"Summenhäufigkeit der Verschleißteilelebensdauer", xlab = "Lebensdauer in h", 
ylab = "relative Häufigkeit")

# Darstellung der Messpunkte als Punkte
points(Breaks.vec, Verschleissteile.df$SummenHfk)
lines(Breaks.vec, Verschleissteile.df$SummenHfk, type = "s", lty = 3)
abline(0.25, 0, lty = "dashed")                # unteres Quartil
abline(0.5, 0, lty = "dashed", col = "red")    # Median
abline(0.75, 0, lty = "dashed")                # oberes Quartil
```



B-2) Erzeuge 500 Zufallszahlen, welche die Lebensdauern von 500 Teilen T simulieren (und somit der Verteilung aus Aufgabe B-1 entsprechen).

```
# Mit der Variable "AnzahlZufallszahlen" gibt man der Funktion mit, wie viele
Zufallszahlen generiert werden sollen.
Zufallszahlen.fct <- function(AnzahlZufallszahlen){
  # Da es sich bei der Datenliste um globale Variablen handelt, kann direkt
auf diese zugegriffen werden, ohne die Werte händisch in einen neuen Vektor
zu schreiben.
  Grenzen.vec <- c(0, Verschleissteile.df$SummenHfk)        # Summenhfk soll
bei 0 starten
  Klassenmitte.vec <- Verschleissteile.df$Klassenmitte     # Aus der Tabelle
(Angaben) Verschleißteile
  Klassenbreite <- Verschleissteile.df[1,2]      # Aus der Tabelle (Angaben)
Verschleißteile

  # Alle vokommenden Intervallsgrenzen
  Break.vec <- c(2750,3250,3750,4250,4750,5250,5750,6250,6750,7250)
  # Erzeugung gleichverteilter Zufallszahlen und anpassen auf die
Anfangsverteilung
  Zufallszahlen.vec <- runif(AnzahlZufallszahlen)
  for(i in 1:AnzahlZufallszahlen){
    for(j in 1:9){
      if((Grenzen.vec[j] < Zufallszahlen.vec[i]) & (Zufallszahlen.vec[i] < Gr
```

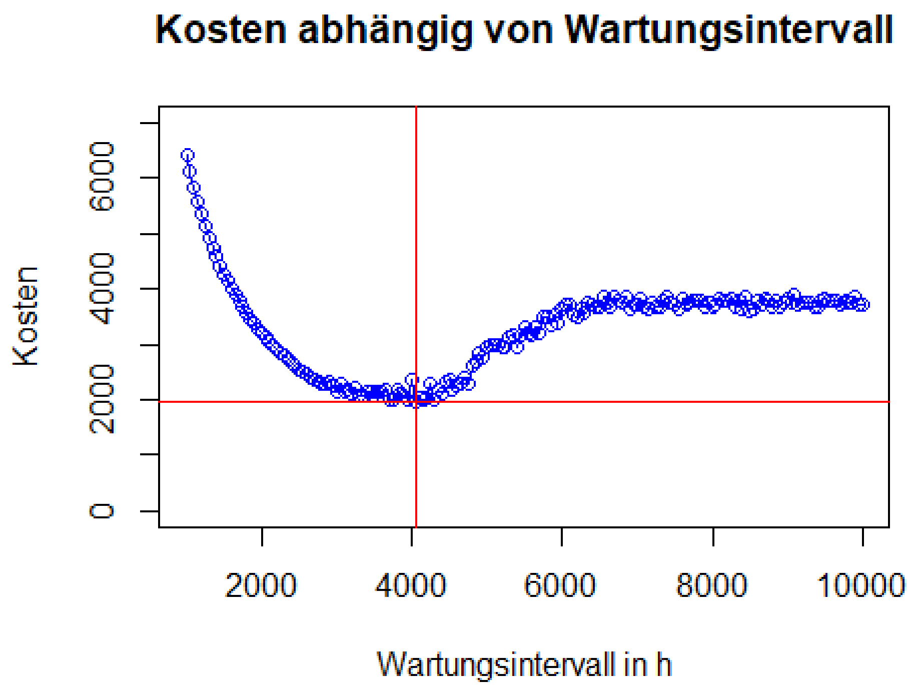
```
enzen.vec[j+1])){
        Zufallszahlen.vec[i] <- (Zufallszahlen.vec[i] - Grenzen.vec[j])/(Gren
zen.vec[j+1] - Grenzen.vec[j])
        Zufallszahlen.vec[i] <- Zufallszahlen.vec[i] * Klassenbreite
        Zufallszahlen.vec[i] <- Zufallszahlen.vec[i] + Break.vec[j]
      }
    }
  }
  return(Zufallszahlen.vec)
}

# Erzeuge 500 Zufallszahlen
Zufallszahlen.vec <- Zufallszahlen.fct(500)

# Darstellen der erzeugten Zufallszahlen zur Kontrolle, ob die Form der
Verteilung in etwa mit der Ausgangsverteilung der "Lebensdauer der Teile"
korreliert
hist(Zufallszahlen.vec, breaks = c(2750,3250,3750,4250,4750,5250,5750,6250,67
50,7250), main = "Verteilung Zufallszahlen Lebensdauer", xlab = "Lebensdauer
in h", right = TRUE, ylab = "absolute Häufigkeit", col = "blue", ylim = c(0,1
50), xlim = c(2000,8000))
```



B-3) Alle Maschinen aus Aufgabe B-1 sind im Betrieb voll ausgelastet. Sowohl Reparaturen als auch ein Teileersatz bei der Wartung führen zu unerwünschten Betriebsunterbrechungen. Die dabei entstehenden Gesamtkosten Kges bestehen aus den Produktionsausfallkosten, den Reparaturkosten und den Materialkosten. Allerdings sind die Gesamtkosten bei Ausfall des Teiles T höher als bei einem planmäßigen Wechsel im Zuge von Wartungsarbeiten: die Betriebsunterbrechung ist länger, die Reparatur dauert länger und der Schadensfall verursacht weitere Folgekosten.

```
# Auflisten der Kosten aus den Angaben
Gesamtkosten.df <- data.frame(Kostenart = c("Betriebsunterbrechnung", "Repara
tur", "Material", "Folgeschäden", "Gesamtkosten"), ErsatzNachAusfall = c(1560
0,1000,800,1200,18600), KostenBeiWartung = c(5200,400,800,0,6400))
Gesamtkosten.df

##                Kostenart ErsatzNachAusfall KostenBeiWartung
## 1 Betriebsunterbrechnung             15600             5200
## 2              Reparatur              1000              400
## 3               Material               800              800
## 4           Folgeschäden              1200                0
## 5           Gesamtkosten             18600             6400

Ausfallskosten.fct <- function(SimZeit, Tau){
  Ausfallskosten <- Gesamtkosten.df[5,2]          # Ermittelt aus Tabelle
Gesamtkosten über Matrixelementzugriff
  Wartungskosten <- Gesamtkosten.df[5,3]          # Ermittelt aus Tabelle
Gesamtkosten über Matrixelementzugriff
  i <- 1          # Laufindex für die Iteration durch die Zeitschleife
  SimPunkt <- 0   # Punkt, an dem die Simulation gerade steht
  AnzahlReparaturAusfall <- 0    # Anzahl der Ausfälle wegen Reparatur
  AnzahlReparaturWartung <- 0    # Anzahl Ausfälle wegen vorab Instandhaltung

  # Simulationszeit um den Faktor 1000 erhöhen - entspicht 1000 h
  while(SimPunkt < (SimZeit*1000)){
    Zufallszahl <- Zufallszahlen.fct(1)   # Generiere genau 1 Zufallszahl
    if(Zufallszahl > Tau){
      AnzahlReparaturWartung <- AnzahlReparaturWartung + 1
      SimPunkt <- SimPunkt + Tau
    } else {
      AnzahlReparaturAusfall <- AnzahlReparaturAusfall + 1
      SimPunkt <- SimPunkt + Zufallszahl
    }

    i <- i + 1
  }
```

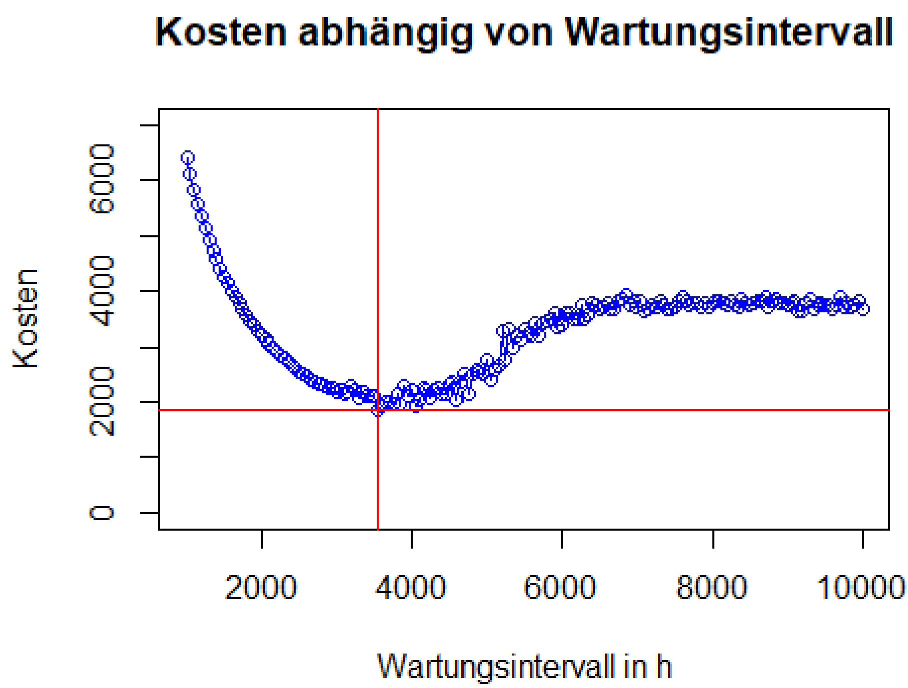
```
  Kges <- (AnzahlReparaturAusfall * Ausfallskosten) + (AnzahlReparaturWartung
* Wartungskosten)              # Berechnen der Gesamtkosten Kges
  return (Kges/SimZeit)         # Funktion hat Gesamtkosten als Rückgabewert
(Entspricht den mittleren Kosten je 1000 Stunden Laufzeit - weil SimZeit mit
Faktor 1000 multipliziert wurde)
}

Simulationszeit <- 500
Kosten.vec <- c()
Tau.vec <- c()

Zeitsprung <- 50            # Schrittweite der Simulationspunkte in h (h Wart
ungsintervall)
minH <- 20                  # Simulationsbeginn bei 1000 h (20*Zeitsprung)
maxH <- 200                 # Ende Simulation bei 10000 h (200*Zeitsprung)


for(n in minH:maxH){
  Kosten.vec <- c(Kosten.vec, Ausfallskosten.fct(Simulationszeit, n*Zeitsprun
g))  # Vektor Kosten.vec bei jeder Iteration um ein Zufallselement erweitern
  Tau.vec <- c(Tau.vec, n*Zeitsprung)    # Vektor Tau.vec bei jeder Iteration
durch die for-Schleife um n*Zeitsprung Element ergänzen
}

plot(Tau.vec, Kosten.vec, ylim = c(0,7000), main = "Kosten abhängig von Wartu
ngsintervall", ylab = "Kosten", xlab = "Wartungsintervall in h", col = "blue"
, type = "o", lty = 1)
# Markierung des Punktes, nach wie vielen Stunden h die Wartung zu den
geringsten Kosten durchgeführt werden kann
abline(h = min(Kosten.vec), v = Tau.vec[which.min(Kosten.vec)], col = "red")
```



In der folgenden Tabelle wird der beste Zeitpunkt für eine Wartung nach h Laufzeit zu den niedrigsten Kosten abgespeichert. Die Tabelle wird mit jeder Iteration der Simulation um die beiden günstigsten Werte ergänzt.

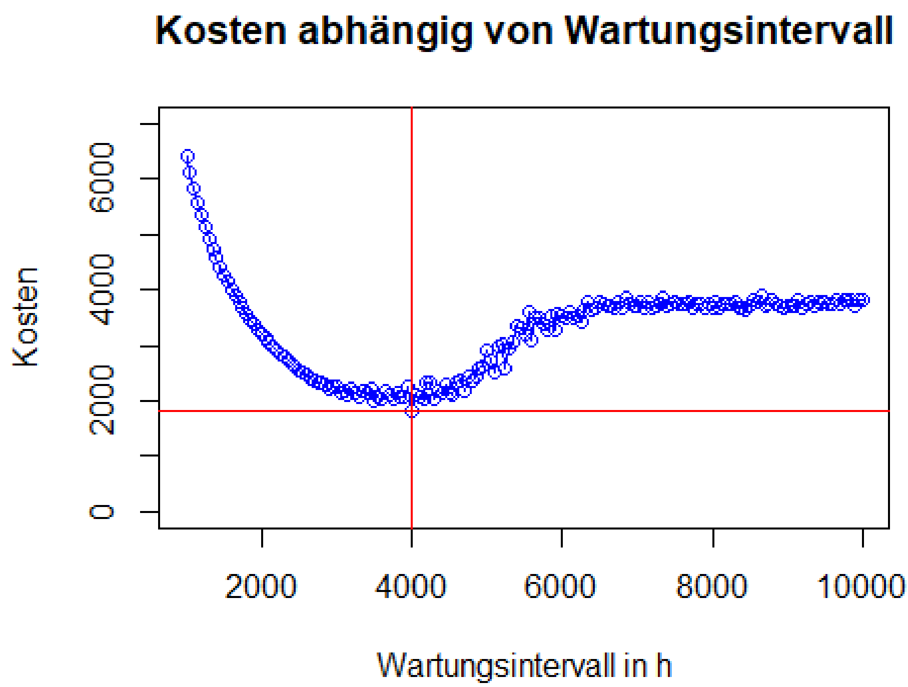
```
XYbesterFall.df <- data.frame(Achsenname = c("Wartungsintervall (x)", "Kosten (y)"), Simulation1 = c((Tau.vec[which.min(Kosten.vec)]), min(Kosten.vec)))
XYbesterFall.df

##              Achsenname Simulation1
## 1 Wartungsintervall (x)      4050.0
## 2            Kosten (y)      1978.8

Simulationszeit <- 500
Kosten.vec <- c()
Tau.vec <- c()

Zeitsprung <- 50    # Schrittweite der Simulationspunkte in h (h Wartungsintervall)
minH <- 20          # Simulationsbeginn bei 1000 h (20*Zeitsprung)
maxH <- 200         # Ende Simulation bei 10000 h (200*Zeitsprung)


for(n in minH:maxH){
  Kosten.vec <- c(Kosten.vec, Ausfallskosten.fct(Simulationszeit, n*Zeitsprung))  # Vektor Kosten.vec bei jeder Iteration um ein Zufallselement erweitern
  Tau.vec <- c(Tau.vec, n*Zeitsprung)     # Vektor Tau.vec bei jeder Iteration durch die for-Schleife um n*Zeitsprung Element ergänzen
```

```
}

plot(Tau.vec, Kosten.vec, ylim = c(0,7000), main = "Kosten abhängig von Wartu
ngsintervall", ylab = "Kosten", xlab = "Wartungsintervall in h", col = "blue"
, type = "o", lty = 1)
# Markierung des Punktes, nach wie vielen Stunden h die Wartung zu den gering
sten Kosten durchgeführt werden kann
abline(h = min(Kosten.vec), v = Tau.vec[which.min(Kosten.vec)], col = "red")
```




```
XYbesterFall.df <- data.frame(XYbesterFall.df, Simulation2 = c((Tau.vec[which
.min(Kosten.vec)]), min(Kosten.vec)))
XYbesterFall.df

##                Achsenname Simulation1 Simulation2
## 1 Wartungsintervall (x)      4050.0      3550.0
## 2               Kosten (y)      1978.8      1866.4

Simulationszeit <- 500
Kosten.vec <- c()
Tau.vec <- c()

Zeitsprung <- 50    # Schrittweite der Simulationspunkte in h (h Wartungsinte
rvall)
minH <- 20          # Simulationsbeginn bei 1000 h (20*Zeitsprung)
maxH <- 200         # Ende Simulation bei 10000 h (200*Zeitsprung)
```

```
for(n in minH:maxH){
  Kosten.vec <- c(Kosten.vec, Ausfallskosten.fct(Simulationszeit, n*Zeitsprun
g))  # Vektor Kosten.vec bei jeder Iteration um ein Zufallselement erweitern
  Tau.vec <- c(Tau.vec, n*Zeitsprung)    # Vektor Tau.vec bei jeder Iteration
durch die for-Schleife um n*Zeitsprung Element ergänzen
}

plot(Tau.vec, Kosten.vec, ylim = c(0,7000), main = "Kosten abhängig von Wartu
ngsintervall", ylab = "Kosten", xlab = "Wartungsintervall in h", col = "blue"
, type = "o", lty = 1)
      # Markierung des Punktes, nach wie vielen Stunden h die Wartung zu den
geringsten Kosten durchgeführt werden kann
abline(h = min(Kosten.vec), v = Tau.vec[which.min(Kosten.vec)], col = "red")
```




```
XYbesterFall.df <- data.frame(XYbesterFall.df, Simulation3 = c((Tau.vec[which
.min(Kosten.vec)]), min(Kosten.vec)))
XYbesterFall.df

##               Achsenname Simulation1 Simulation2 Simulation3
## 1 Wartungsintervall (x)      4050.0      3550.0        4000
## 2             Kosten (y)      1978.8      1866.4        1808
```

Bei dreimaliger Simulation zeigt sich, dass durch die verwendeten Zufallszahlen die Werte für die x-Komponente (Wartungsintervall in h) und die Werte für die y-Komponente (Kosten) einer gewissen Schwankung unterliegen. Über die folgende Mittelwertsfunktion kann der Mittelwert dieser Schwankung ermittelt werden. Auch wenn die Stichprobe aus 3 Elementen sehr gering ist, kann dennoch eine Schwankung festgestellt werden.

```
MittelwertKosten <- ((XYbesterFall.df[2,2])+(XYbesterFall.df[2,3])+(XYbesterF
all.df[2,4]))/3
MittelwertKosten

## [1] 1884.4

MittelwertWartung <- ((XYbesterFall.df[1,2])+(XYbesterFall.df[1,3])+(XYbester
Fall.df[1,4]))/3
MittelwertWartung

## [1] 3866.667
```

Aus den Wert für den Wartungsintervall bei den besten Kosten kann dann berechnet werden, nach wie viel Prozent der mittleren Lebensdauer der optimal Wartungszeitpunkt zu den geringsten Kosten ist.

```
BesterZeitpunkt <- MittelwertWartung/Erwartungswert*100
BesterZeitpunkt

## [1] 78.11448
```

Der beste Wartungszeitpunkt zu den geringsten Kosten ist somit bei 78.1144781 % der mittleren/erwarteten Lebensdauer.