Die Monte Carlo (MC) Methode

1. Idee:

Sei $F(x)$ eine beliebige Verteilungsfunktion und existiere der Erwartungswert einer Funktion $g(X)$, d.h. $\mathsf{E}(g(X)) = \int g(x)dF(x) < \infty$. Dann gilt für $X^{(1)}, \ldots, X^{(R)} \overset{iid}{\sim} F(x)$ (Starkes Gesetz der großen Zahlen)

$$\widehat{\mathsf{E}}_{\mathsf{MC}}\big(g(X)\big) = \frac{1}{R}\sum_{r=1}^{R} g\big(X^{(r)}\big) \overset{f.s.}{\to} \mathsf{E}\big(g(X)\big)\,.$$

R nennt man *Replikationszahl*. Der Monte-Carlo Schätzer $\widehat{\mathsf{E}}_{\mathsf{MC}}(g(X))$ ist selbst eine Zufallsvariable, die für $R \to \infty$ fast sicher gegen den gesuchten wahren Erwartungswert $\mathsf{E}(g(X))$ strebt.

Allgemeiner MC Algorithmus in R:

```
 n <- 1      # sample size, dim(X)=1
 R <- 1000   # number of replications
 z <- 1:R    # initialize z as a list of R elements
 for (r in 1:R) {
   x <- rF(n, par)
   z[r] <- g(x)
}
mean(z)
```

Verfügbare Zufallszahlengeneratoren `rF` in `R` mit Defaultwerten für die Parameter:

- N(μ, σ^2): `rnorm(n, mean=0, sd=1)`
- Uniform$(\min, \max)$: `runif(n, min=0, max=1)`
- Beta(a, b): `rbeta(n, a, b)`
- Binom(s, p): `rbinom(n, size, prob)`
- Cauchy(α, σ): `rcauchy(n, loc=0, scale=1)`
- $\chi^2(df, ncp)$: `rchisq(n, df, ncp = 0)`, (entspricht Gamma$(df/2, 1/2)$)
- Exp$(rate)$: `rexp(n, rate=1)`
- F(n_1, n_2): `rf(n, df1, df2)`

- Gamma(a, s): `rgamma(n, shape, rate=1, scale=1/rate)`
- Geom(p): `rgeom(n, prob)`
- Hyper(m, n, k): `rhyper(nn, m, n, k)`
- LogN(μ, σ^2): `rlnorm(n, meanl=0, sdl=1)`
- Logistic(μ, σ^2): `rlogis(n, loc=0, scale=1)`
- NegBinom(s, p): `rnbinom(n, size, prob, mu)`
- Poisson(λ): `rpois(n, lambda)`
- t(df): `rt(n, df)`
- Weibull(a, b): `rweibull(n, shape, scale=1)`

`R` bietet auch Funktionen zur Berechnung der Dichte/Wahrscheinlichkeitsfunktion (`dF`), Verteilungsfunktion (`pF`) und Quantilsfunktion (`qF`), wobei `F` so wie zuvor bei den Generatoren definiert ist.

Anwendung: Sei $X_n = X_1, \ldots, X_n$ eine n-elementige Zufallsstichprobe aus F. Untersuche die Varianzen von $\overline{X}_n$ und $\tilde{X}_n$ für endliche (**kleine**) $n < \infty$. Berechne dazu die **Monte Carlo Schätzer** für $\mathsf{var}(\tilde{X}_n)$, $\mathsf{var}(\overline{X}_n)$, und für die Asymptotische Relative Effizienz

$$\mathsf{are}(\overline{X}_n, \tilde{X}_n) = \frac{\mathsf{var}(\tilde{X}_n)}{\mathsf{var}(\overline{X}_n)}\,.$$

Z.B. für $\tilde{X}_n$:

$$\mathsf{var}(\tilde{X}_n) = \int \big(x - \mathsf{E}(\tilde{X}_n)\big)^2 dF_{\tilde{X}_n}(x)\,.$$

Was auch immer die exakte Verteilungsfunktion $F_{\tilde{X}_n}$ des empirischen Medians einer n-elementigen Zufalls-Stichprobe aus F sein mag, wir benötigen nur recht viele Replikationen von $\tilde{X}_n$ aus $F_{\tilde{X}_n}$.

Seien diese $\tilde{X}_n^{(1)}, \ldots, \tilde{X}_n^{(R)} \overset{iid}{\sim} F_{\tilde{X}_n}$, dann gilt

$$\widehat{\mathsf{var}}_{\mathsf{MC}}(\tilde{X}_n) = \frac{1}{R} \sum_{r=1}^{R} \left(\tilde{X}_n^{(r)} - \widehat{\mathsf{E}}_{\mathsf{MC}}(\tilde{X}_n) \right)^2 \overset{f.s.}{\to} \mathsf{var}(\tilde{X}_n)$$

mit

$$\widehat{\mathsf{E}}_{\mathsf{MC}}(\tilde{X}_n) = \frac{1}{R} \sum_{r=1}^{R} \tilde{X}_n^{(r)} \overset{f.s.}{\to} \mathsf{E}(\tilde{X}_n)\,.$$

R soll dabei groß gewählt sein, so dass der MC Schätzer stabil ist. Als Faustregel verwendet man zumindest $100 < R < 1000$ für Momente und $R > 1000$ für Quantile x_α. Je größer oder kleiner das Niveau des Quantils α ist, d.h. je näher α bei 0 oder 1 liegt, desto größer muss R gewählt werden.

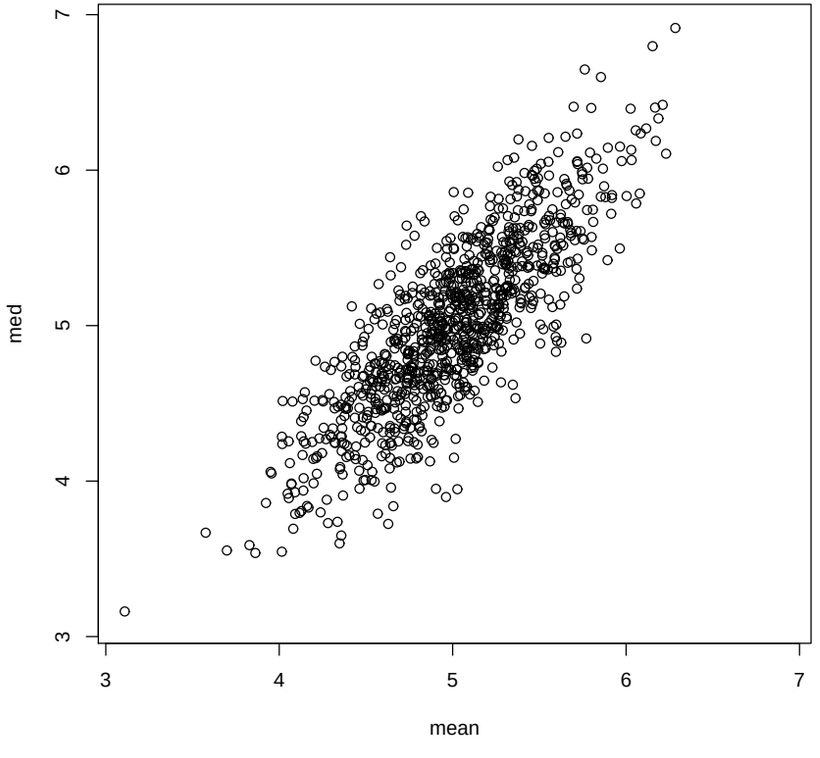
MC Algorithmus in R:

```
n <- 20       # sample size, dim(X_n)=n
R <- 1000     # number of replications
med <- 1:R    # initializations
mean <- 1:R
for (r in 1:R) {
  x <- rF(n, par)
  med[r] <- median(x)
  mean[r] <- mean(x)
}
areMC <- var(med) / var(mean)
```

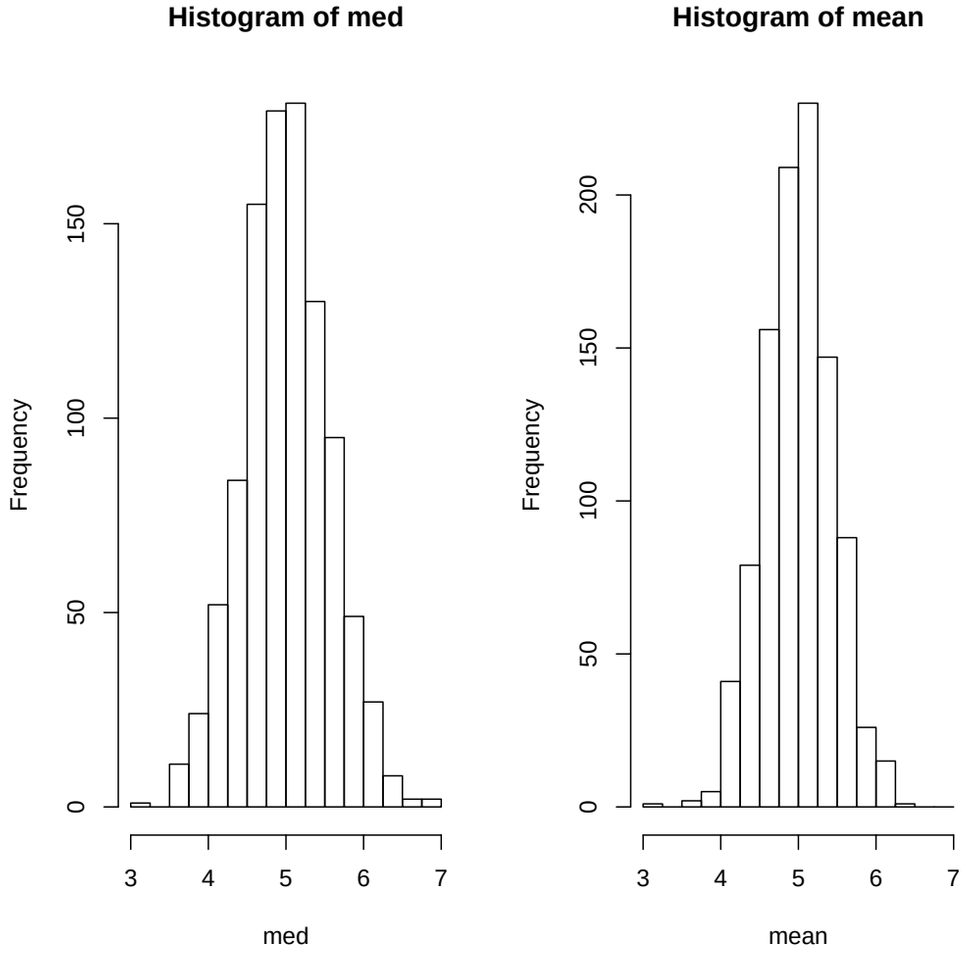
Zur Erinnerung gilt asymptotisch für beliebig normalverteilte Zufalls-Stichproben $\lim_{n\to\infty} \text{are}(\overline{X}_n, \tilde{X}_n) = \pi/2$. Der MC Schätzer erlaubt auch speziell für kleine Werte von n eine Aussage.

Verwendet man für `rF(n, par)` speziell `rnorm(n, 5, 2)`, so führt dies (zufälligerweise) zu Monte-Carlo Realisationen $(\tilde{x}_n^{(r)}, \overline{x}_n^{(r)})$ mit folgender Struktur:

```
plot(mean, med)
```



```
hist(med, xlim=c(3, 7)); hist(mean, xlim=c(3, 7))
```



MC Methode wird auch zur Überprüfung der **Überdeckungswahrscheinlichkeit** $1-\alpha$ eines Konfidenzintervalls für θ genutzt. Sei dazu $(L^{(r)}, U^{(r)})$, $r = 1, \ldots, R$, eine Folge von iid Konfidenzintervalle für θ zum Niveau $1-\alpha$.

Generiere R mal Zufallsstichprobe mit Umfang n aus $F(\theta_0)$ (θ_0 ist wahre Parameter), und berechne die r-te Realisation des Konfidenzintervalls. Dann gilt

$$1 - \widehat{\alpha}_{\mathsf{MC}} = \frac{1}{R} \sum_{r=1}^{R} I_{[L^{(r)}, U^{(r)}]}(\theta_0) \overset{f.s.}{\to} 1 - \alpha \,.$$

Für Zufallsstichprobe aus $N(\mu, \sigma^2)$-Verteilung (mit σ^2 bekannt) liefert das zweiseitige Konfidenzintervall für μ

$$\overline{X}_n \pm z_{1-\alpha/2} \sigma / \sqrt{n}$$

bekanterweise eine Überdeckungswahrscheinlichkeit von $1-\alpha$.

```
n <- 20       # sample size
R <- 1000     # number of replications
mu <- 5;  sigma <- 2 # true parameter(s)
alpha <- 0.05         # 1 - coverage probability
L <- U <- 1:R;  a <- sigma/sqrt(n) * qnorm(1 - alpha/2)
for (r in 1:R) {
  m <- mean(rnorm(n, mu, sigma))
  L[r] <- m - a;   U[r] <- m + a
}
left  <- as.numeric(mu < L); sum(left)
[1] 27
right <- as.numeric(U < mu); sum(right)
[1] 25
```

Wahrer Parameter ($\mu_0 = 5$) liegt 27 mal unter der unteren und 25 mal über der oberen Grenze, d.h. in 52 (von 1000) Fällen wird μ_0 nicht überdeckt, was einem MC Schätzer $\widehat{\alpha}_{\mathsf{MC}} = 0.052$ (bei vorgegebenem $\alpha = 0.05$) entspricht.

2. Bootstrap/Stiefelriemen/Münchhausen-Trick

Bis jetzt: vollständig spezifiziertes Modell $F(\theta)$ für MC Simulation angenommen, kein Bezug zu einer konkreten Datensituation.

Jetzt: Sei $X_1, \ldots, X_n$ Stichprobe aus unbekanntem Modell F. Zwar kennen wir F nicht, haben aber daraus eine Stichprobe vom Umfang n. Beim Bootstrap wird nun die Stichprobeninformation auf zweierlei Art verwendet.

Parametrischer Bootstrap: wie zuvor Verteilung $F(\theta)$ für Stichprobe annehmen. Parameter θ durch $Schätzer\, \hat{\theta}$ aus Stichprobe ersetzen. Nimmt man beispielsweise an, dass $X_1, \ldots, X_n \sim N(\mu, \sigma^2)$, so basiert der parametrische Bootstrap auf die generierte Stichprobe (eine Replikation) $X_1^*, \ldots, X_n^*$ mit $X_i^* \sim N(\overline{x}, s^2)$.

Nicht-Parametrischer Bootstrap: verzichtet gänzlich auf Verteilungsannahme und verwendet die $empirische\ Verteilungsfunktion$ als Schätzer für F. Die Replikation kommt somit aus $\hat{F}_n$. Realisiert wird dieses Verfahren, indem n mal **mit Zurücklegen** $X_1^*, \ldots, X_n^*$ aus der Realisierung $x_1, \ldots, x_n$ gezogen wird.

Beide Ansätze basieren auf $X_1^*, \ldots, X_n^*$ aus der geschätzten Verteilungsfunktion. Ist man z.B. an der Schätzung der Varianz des Medians interessiert, also an $\text{var}(\tilde{X}|X_i \sim F)$, so liefern die Bootstrap-Schätzer $\text{var}(\tilde{X}^*|X_i^* \sim F(\hat{\theta}))$ oder $\text{var}(\tilde{X}^*|X_i^* \sim \hat{F}_n)$. Nur selten sind Bootstrap-Momente analytisch berechenbar. Daher wiederum MC-Methode verwendet.

Allgemeiner MC-Bootstrap Algorithmus in `R`:

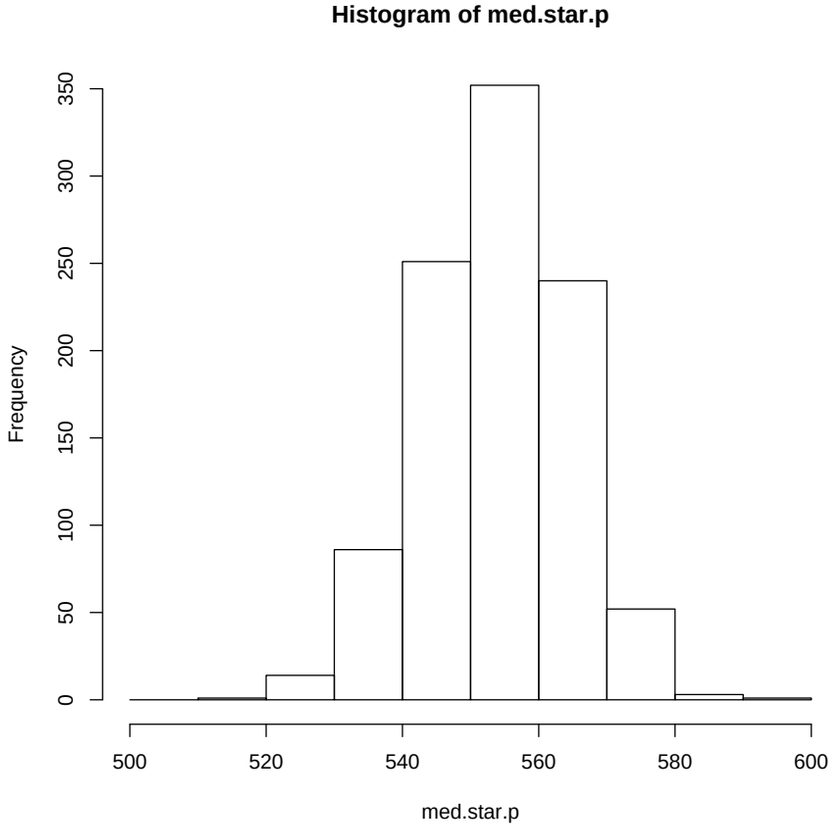
```
n <- length(x);  R <- 1000
med.star <- 1:R
for (r in 1:R) {
  x.star <- rF(n, par.estimate)           # parametric Bootstrap
  x.star <- sample(x, size=n, replace=T) # non-param. Bootstrap
  med.star[r] <- median(x.star)
}
EMC.median <- mean(med.star)
varMC.median <- var(med.star)
```

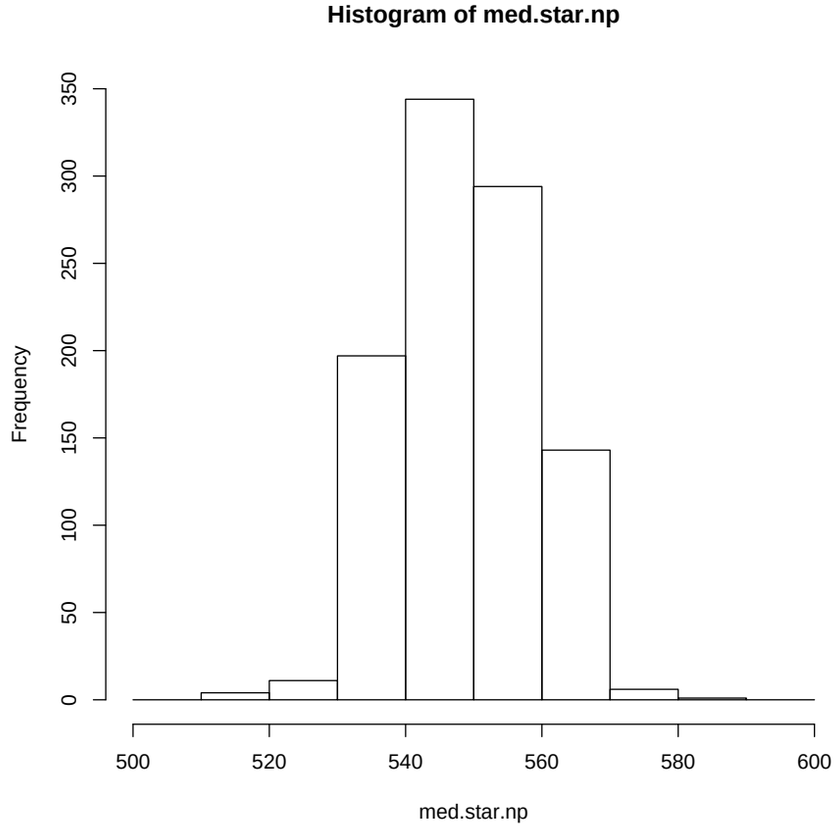
Als MC Approximation der Bootstrap-Schätzung vom ARE(Median, Mean) erhält man unter Normalverteilungsannahme für die Variable `fvc` aus Datensatz `aimu`

```
aimu <- read.table("aimu.dat")
attach(aimu)
n <- length(fvc)
R <- 1000
med.star.p <- mean.star.p <- med.star.np <- mean.star.np <- 1:R
for (r in 1:R) {
  x.star.p <- rnorm(n, mean(fvc), sd(fvc))    # parametric BT
  x.star.np <- sample(fvc, size=n, replace=T) # non-param. BT
  mean.star.p[r] <- mean(x.star.p)
  med.star.p[r] <- median(x.star.p)
  mean.star.np[r] <- mean(x.star.np)
  med.star.np[r] <- median(x.star.np)
}
```

```
 are.MCB.p <- var(med.star.p)/var(mean.star.p); are.MCB.p
 [1] 1.523219
 are.MCB.np <- var(med.star.np)/var(mean.star.np); are.MCB.np
 [1] 1.369213

breaks <- seq(from=500, to=600, by=10)
hist(med.star.p, breaks, xlim=c(500, 600), ylim=c(0,350))
hist(med.star.np, breaks, xlim=c(500, 600), ylim=c(0,350))
```


Histogram of med.star.p
Frequency
0
50
100
150
200
250
300
350
500
520
540
560
580
600
med.star.p


Histogram of med.star.np
Frequency
0
50
100
150
200
250
300
350
500
520
540
560
580
600
med.star.np