Zentrale Grenzwertsatz

Zentrale Grenzwertsatz

Für jedes $n = 1, 2, \ldots$ seien die Zufallsvariablen $X_1, X_2, \ldots, X_n$ unabhängig und besitzen die gleiche Verteilung mit dem Erwartungswert $\mu = E(X_i)$ und der Varianz $\sigma^2 = Var(X_i)$. Dann gilt für die Verteilungsfunktion der standardisierten Summen $G_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma}$:

$$\lim_{n\to\infty} P\left(G_n \leq x\right) = \Phi(x)$$

Dabei ist Φ die Verteilungsfunktion der Standard-Normalverteilung $N(0, 1)$.

Überprüfen Sie mit R den zentralen Grenzwertsatz.

Verteilung

Erzeugen Sie einen Data Frame, der die gezogenen Zufallszahlen G_n enthält. Die Stichprobe soll einen Umfang von 50000 haben (Anzahl Zeilen der Tabelle). Der Data Frame hat sechs Spalten, die die Werte für die Zahlen $n = 1, 2, 3, 10, 100, 1000$ enthalten. Geben Sie die ersten 10 Zeilen des Data Frames aus.

Welche Verteilung Sie für X_i nehmen, bleibt Ihnen überlassen. Es soll nur keine Normalverteilung sein.

```
# Erwartungswert der Chi-Quadrat Verteilung ist n (Anzahl der Freiheitsgrade).
# Standardabweichung ist \sqrt(2*n)
library(stats)
set.seed(1)
standardized_sum = function(vec, mu, sd){
  n = length(vec)
  (sum(vec)- n*mu) / (sqrt(n)*sd)
}
iter <- 1
list_n_1 <- list()
list_n_2 <- list()
list_n_3 <- list()
list_n_10 <- list()
list_n_100 <- list()
list_n_1000 <- list()
while (iter <= 50000){
  list_n_1[[iter]] <- rchisq(1,2)
  list_n_2[[iter]] <- rchisq(2,2)
  list_n_3[[iter]] <- rchisq(3,2)
  list_n_10[[iter]] <- rchisq(10,2)
  list_n_100[[iter]] <- rchisq(100,2)
  list_n_1000[[iter]] <- rchisq(1000,2)
  iter = iter + 1

}
chi_n_1  = sapply(list_n_1, standardized_sum, mu=2, sd=sqrt(2*2))
chi_n_2  = sapply(list_n_2, standardized_sum, mu=2, sd=sqrt(2*2))
```

```r
chi_n_3  = sapply(list_n_3, standardized_sum, mu=2, sd=sqrt(2*2))
chi_n_10  = sapply(list_n_10, standardized_sum, mu=2, sd=sqrt(2*2))
chi_n_100 = sapply(list_n_100, standardized_sum, mu=2, sd=sqrt(2*2))
chi_n_1000  = sapply(list_n_1000, standardized_sum, mu=2, sd=sqrt(2*2))


df = data.frame(chi_n_1,chi_n_2,chi_n_3,chi_n_10,chi_n_100,chi_n_1000)
head(df,n=10)
```

```
##       chi_n_1     chi_n_2     chi_n_3    chi_n_10   chi_n_100 chi_n_1000
## 1  -0.8448586  1.19282839  0.12532920 -1.26006741 -0.19350511  0.4415122
## 2  -0.6293512 -0.72408987  0.01530347 -0.96961016  0.02297818  1.5065334
## 3  -0.9793138  0.10522989  2.02770501  1.72259012 -0.91369892 -0.7437185
## 4  -0.7149816 -0.52377355 -0.76685819  0.66180366 -0.25252525  0.6859781
## 5   2.4376516 -1.17980219 -1.04677051 -0.63974014 -0.51230427  1.6671277
## 6   0.1190443  0.21464318  0.29801712  0.44085283 -0.73575882 -0.8684604
## 7   2.1183481 -0.01419154  1.67340914 -0.51935835 -0.19399917  0.6932579
## 8  -0.3350435 -0.85341453 -0.33333033 -1.00187868 -0.08948624 -0.3697799
## 9  -0.6611039 -1.35604313  0.71771453  0.02504821  0.22588269  1.5910629
## 10 -0.4521271  1.37780897 -0.69495649  0.94941848  0.14203376  0.5201319
```

Plot

Plotten Sie sechs Histogramme, die je für $n = 1, 2, 3, 10, 100, 1000$ die Verteilung im Vergleich zu einer $(0, 1)$-Normalverteilung zeigen. Die Intervallbreite soll 0,25 sein.

Tipp: Platzieren Sie die sechs Plots auf einem 3×2-Gitter.

```r
# Ihre Lösung:
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

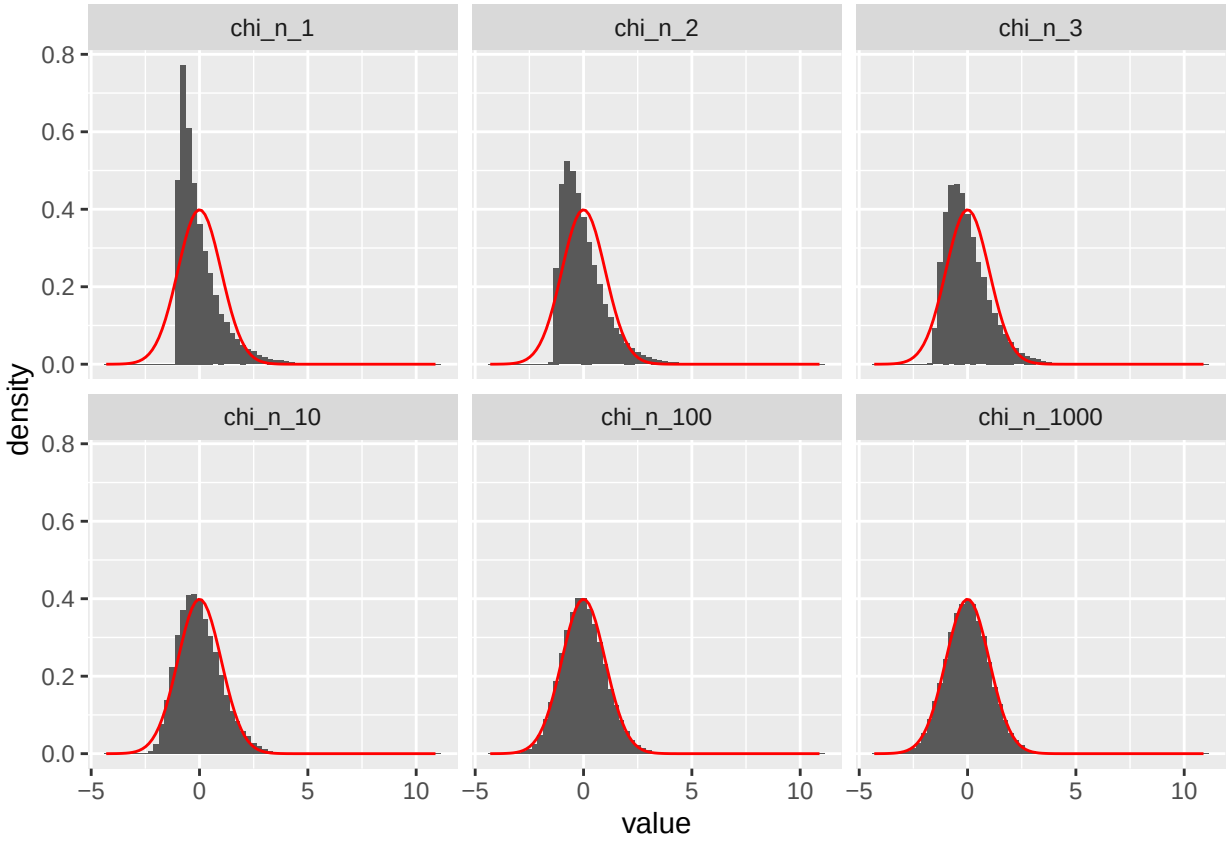
```r
library(ggplot2)

df_long = pivot_longer(df, cols=everything(), names_to="n", values_to="value")
df_long$n <- factor(df_long$n, levels= c("chi_n_1", "chi_n_2", "chi_n_3",
             "chi_n_10", "chi_n_100", "chi_n_1000"))
normal = dnorm(1)

#ggplot(normal, aes(x=value))+
 # geom_histogram(bins=50)

ggplot(df_long, aes(x=value))+
  geom_histogram(aes(y=after_stat(density)),binwidth=0.25)+
  # Histogramme müssen auf density gestellt werden,
  # damit Normalverteilung eingezeichnet werden kann
```

```
  facet_wrap(~n, ncol=3)+
  stat_function(fun=dnorm, color="red")
```



Abweichung

Nun soll die Abweichung der Standardisierten G von der Normalverteilung für alle n quantifiziert werden. Hierzu soll für alle Balken (bins) der Histogramme aus der vorigen Aufgabe die Differenz von G zur Normalverteilung gebildet und quadriert werden. Diese Werte werden aufaddiert und durch die Anzahl der Intervalle geteilt. Daraus wird die Wurzel gezogen.

Sie können sich auch ein anderes Maß zur Bestimmung der Abweichung überlegen.

Geben Sie die Abweichungen aus. Stimmt es, dass die Abweichungen mit größerem n kleiner werden?

Tipp: `hist(plot = FALSE)` erzeugt ein Histogramm, ohne es zu plotten. Gerne können Sie auch Ihr eigenes Histogramm nutzen.

```
sims <- list(
  "1" = chi_n_1,
  "2" = chi_n_2,
  "3" = chi_n_3,
  "10" = chi_n_10,
  "100" = chi_n_100,
  "1000" = chi_n_1000
)
diff_norm = function(G){
  # Intervallbreiten der vorigen Aufgabe generieren (0.25er Schritte)
  breaks <- seq(floor(min(G)), ceiling(max(G)) + 0.25, by = 0.25)
```

```r
  h <- hist(G, breaks = breaks, plot=FALSE)  # errechne Histogramm ohne zu plotten
  sqrt(mean((h$density- dnorm(h$mids))^2)) # berechnet die Abweichung
}

sapply(sims, diff_norm)
```

```
##           1           2           3          10         100        1000
## 0.110211417 0.064618054 0.051631678 0.024998170 0.007589740 0.003755681
```

Ja die Abweichungen werden mit größerem n kleiner.