# TDAB01 SANNOLIKHETSLÄRA OCH STATISTIK

# LABB 1: SIMULERING

JOSE M. PEÑA, MÅNS MAGNUSSON, MATTIAS VILLANI
IDA, LINKÖPINGS UNIVERSITET

## 1. INSTRUKTIONER

- Laborationen ska göras **två och två**
- Labben ska vara en **PDF-rapport** med kod, analys och grafer. I rapporten ska följande ingå:
  - Båda studenternas namn och LiU-id.
  - Laborationsnummer
  - Uppgifterna ni besvarar (t ex som rubriker).
- Ett tips är att använda R markdown. Här finns en R markdownmall att utgå ifrån. Antingen kan ni skapa PDF direkt från denna mall i R-Studio (med TeX) eller så skapar ni ett Word/HTML dokument som sedan skrivs ut till PDF.
- Laborationsrapporten skickas in via LISAM. Där hittar ni också **deadlinen**.

## 2. INTRODUKTION TILL R

R är ett programmeringspråk för statistisk programmering som påminner mycket om Matlab. R bygger på öppen källkod och kan laddas ned här. R-Studio är en mycket populär IDE för R (som också påminner mycket om Matlab). Denna IDE finns att tillgå här. I R-Studio finns funktionalitet för literate programming med R markdown implementerat för att kombinera R kod med markdownsyntax. På detta sätt är det enkelt att generera rapporter med både text, grafik och kod. Det är R:s motsvarighet till Python Notebook.

För en ingång till R från andra språk kan onlineboken *Advanced R* rekommenderas som finns här. Kapitlen *Data structures*, *Subsetting* och *Functions* bör ge en snabb introduktion.

Även boken *The art of R programming* av Norman Matloff kan vara till hjälp som referenslitteratur. Boken finns här.

### 2.1. **Videomaterial.**

- För en introduktion till syntaxen i R se Google developers R videomaterial här.
- Mer (detaljerat) videomaterial av Roger Peng finns att tillgå här.
- För att visualisera med basgrafiken finns följande introduktionsvideo.
- För mer komplicerad grafik rekommenderas `ggplot2` paketet. En introduktionsvideo finns här.
- En introduktion till R markdown finns här.

### 2.2. **Cheatsheets.**

- *R reference card v.2* av Matt Baggot med vanliga funktioner i R finns att tillgå här.
- *R markdown cheatsheet* av R-Studio med tips för R markdown finns att tillgå här.

## 3. LABORATION

3.1. **Simulering och Monte Carlo metoder.** De första uppgifterna i denna laboration syftar till hur vi kan använda oss av simulering (eller Monte Carlo metoder) för att studera sannolikhetsfördelningar och lösa statistiska problem. I R är det mycket enkelt att simulera från de flesta sannolikhetsfördelningar som har behandlats i kursen såhär långt.

För att dra 5 slumpmässiga värden från $\mathcal{N}(\mu = 3, \sigma = 1)$ gör vi på följande sätt:

```
> x <- rnorm(n = 5, mean = 3, sd = 1)
> x

[1] 2.78434 3.30187 1.04399 0.69534 3.90852
```

Funktionen returnerar en vektor med utfall från vår fördelning. På liknande sätt kan 3 värden från exponentialfördelningen dras på följande sätt:

```
> y <- rexp(n = 3, rate = 10)
> y

[1] 0.040121 0.038042 0.059220
```

. . . och slumpmässiga värden från binomialfördelningen dras på följande sätt:

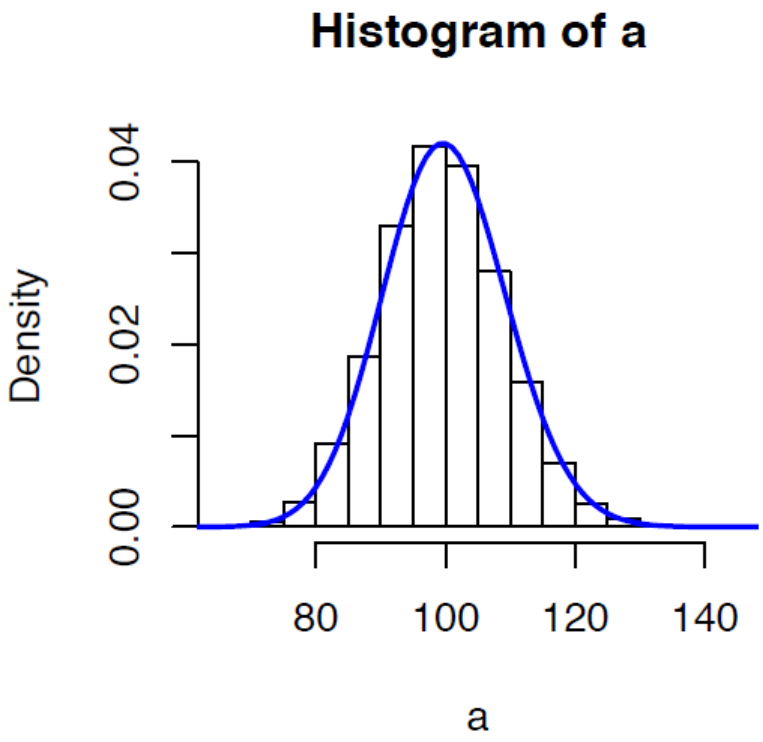
```
> z <- rbinom(n = 4, size = 1000, prob = 0.1)
> z

[1] 116 116 102 104
```

Observera att `n` i funktionen anger hur många dragningar från fördelningen du vill ha. I binomialfördelningen är det vanligt att parametrisera fördelningen som Bin$(n, p)$. I R motsvaras $n$ av `size` och $p$ av `prob`.

För att enkelt visualisera en fördelning kan vi använda funktionen `hist()` och fördelningens täthetsfunktion på följande sätt:

```
> a <- rbinom(n = 10000, size = 1000, prob = 0.1)
> hist(a, probability = TRUE)
> xfit <- seq(50, 150, 1)
> yfit <- dbinom(xfit, size = 1000, prob = 0.1)
> lines(xfit, yfit, col="blue", lwd=2)
```



3.1.1. *Simulering av normalfördelning.* Simulera 100 och 10000 dragningar från en normalfördelning med $\mu = 10$ och $\sigma = 4$.

(1) Visualisera fördelningarna i två histogram. Visualisera fördelningens pdf i samma graf.
(2) Beskriv skillnaden mellan de olika graferna.

3.1.2. *Simulera och visualisera andra fördelningar.*

(1) Simulera och visualisera följande fördelningar med 10000 dragningar från varje fördelning samt fördelningens täthetsfunktioner.

**Diskreta fördelningar**

$$\begin{aligned} X_1 &\sim \text{Bernoulli}(p = 0.2) \\ X_2 &\sim \text{Binomial}(n = 20, p = 0.1) \\ X_3 &\sim \text{Binomial}(n = 20, p = 0.5) \\ X_4 &\sim \text{Geometrisk}(p = 0.1) \\ X_5 &\sim \text{Poisson}(\lambda = 10) \end{aligned}$$

**Kontinuerliga fördelningar**

$$\begin{aligned} Y_1 &\sim \text{Likformig}(\text{min} = 0, \text{max} = 1) \\ Y_2 &\sim \text{Exp}(\theta = 3) \\ Y_3 &\sim \text{Gamma}(\alpha = 2, \beta = 1) \\ Y_4 &\sim \text{Student-t}(\nu = 3) \\ Y_5 &\sim \text{Beta}(\alpha = 0.1, \beta = 0.1) \\ Y_6 &\sim \text{Beta}(\alpha = 1, \beta = 1) \\ Y_7 &\sim \text{Beta}(\alpha = 10, \beta = 5) \end{aligned}$$

**Tips!** $\alpha$ heter `shape` och $\beta$ heter `scale` för gammafördelningen i R.

Som behandlats under föreläsningarna finns det också relationer mellan olika fördelningar. Vi kunde se det i föregående uppgift där betafördelningen med $\alpha = \beta = 1$ ger en likformig fördelning, $U(0, 1)$. Det finns flera sådana relationer som är värdefulla att känna till.

3.1.3. *Relation mellan fördelningar.* Nedan finns två stycken sannolikhetsfördelningar som konvergerar mot andra sannolikhetsfördelningar.

$$\begin{aligned} X &\sim \text{Binomial}(n = 10000, p = 0.001) \\ Y &\sim \text{Student-t}(\nu = 10000) \end{aligned}$$

(1) Simulera 1000 värden från respektive fördelning och visualisera fördelningen i ett histogram tillsammans med fördelningens täthetsfunktionen.
(2) Ta reda på (t ex via Wikipedia eller föreläsningarnas slides) vilken annan fördelning som respektive fördelning börjar konvergera mot.
(3) Simulera dragningar från dessa fördelning och jämför resultatet med de resultat du fick i (1).

Som vi har sett är det mycket enkelt att simulera värden från olika fördelningar i R. Att simulera värden från en fördelning är ofta ett alternativ till att lösa sannolikhetsproblem analytiskt, särskilt om den analytiska lösningen är komplicerad eller omöjlig att beräkna.

3.1.4. *Analytisk sannolikhet och approximation med Monte Carlo metoder.* Varje fördelning i R har också en funktion för att beräkna sannolikheten direkt. På samma sätt som funktioner simulerar värden med prefixet `r` har vi den kumulativa fördelningsfunktionen med prefixet `p` och täthetfunktionen med prefixet `d`.
I denna uppgift ska vi arbeta med två fördelningar

$$\begin{aligned} X &\sim \mathcal{N}(\mu = 0, \sigma = 1) \\ Y &\sim \text{Bin(n=10}, p = 0.1) \end{aligned}$$

(1) Använd täthetsfunktionen i R för att beräkna $\boldsymbol{P}(Y = 0)$. Simulera nu 10000 dragningar från $Y$ och beräkna andelen dragningar där $y = 0$.
(2) Använd nu den kumulativa fördelningsfunktionen i R för att beräkna följande sannolikheter.
- $\boldsymbol{P}(X < 0)$
- $\boldsymbol{P}(-1 < X < 1)$
- $\boldsymbol{P}(1.96 < X)$
- $\boldsymbol{P}(0 < Y < 10)$
- $\boldsymbol{P}(Y = 0)$
- $\boldsymbol{P}(0 \leq Y \leq 10)$

(3) Beräkna samma sannolikheter som i (2) men genom att simulera dragningar från $X$ och $Y$ i R.

Som ett nästa steg ska vi försöka göra en lite mer realistisk beräkning av sannolikheter och hur simuleringar kan användas för att hjälpa oss fatta beslut under osäkerhet. Vi kan, genom att specificera vår osäkerhet som sannolikhetsfördelningar, enkelt få svar på komplexa frågor genom simuleringar.

3.1.5. *Beräkna (icke-triviala) sannolikheter.* Du är CTO för ett företag och är osäker på om du ska bygga om ert nuvarande system för att minska kostnaden för underhåll. I ert nuvarande system vet du att ni har en felfrekvens cirka 10%, eller $p = 0.1$, sett till antalet kunder och år. Ni har totalt $337$ kunder och antalet fel/buggar nästa år kan således beskrivas som en Bin$(n = 337, p = 0.1)$.
Det alternativ du står inför är betydligt mycket osäkrare, du vet inte hur stor felfrekvensen kommer bli så du uppskattar att felfrekvensen, $p$, kommer ligga mellan 0.02 och 0.16. Din osäkerhet kan således beskrivas som en likformig fördelning $P \sim U(0.02, 0.16)$, baserat på dessa dragningar följer sedan antalet fel Bin$(n = 337, p = P)$.
För att lösa detta problem, gör 10 000 simuleringar för det nuvarande systemet och 10 000 simuleringar för det gamla systemet och besvara följande frågor. **Obs!** Tänk på att en sannolikhet är definierad på intervallet $[0, 1]$.
(1) Vad är det förväntade antalet fel för de båda systemen?
(2) Vad är sannolikheten att det nuvarande systemet genererar färre fel än det nya systemet?
(3) Vad är sannolikheten att du kommer få fler än 50 fel i respektive system?

3.2. **Stora talens lag.** En del av de resultat vi fick ovan bygger delvis på stora talens lag vilket innebär att

$$\lim_{n\to\infty} \boldsymbol{P}(|\bar{X}_n - \mu| > \epsilon) = 0$$

där $\bar{X}_n = \frac{1}{n}\sum_i^n X_i$. Uttryckt annorlunda kommer medelvärdet av $n$ slumpmässiga variabler konvergera mot det teoretiska förväntade värdet, $\mu$, av fördelningen.

3.2.1. *Stora talens lag.* Vi ska studera följande två stokastiska variabler

$$X \sim \text{Bin}(n = 10, p = 0.2)$$
$$Y \sim \text{Gamma}(\alpha = 2, \beta = 2)$$

(1) Använd föreläsningarnas slides eller Wikipedia för att beräkna de (teoretiska) förväntade värdena ($E(X)$ och $E(Y)$) för dessa stokastiska variabler.
(2) Dra 10, 20, ... 90, 100, 200, ..., 900, 1000, 2000, ..., 9000 och 10000 antal värden från respektive fördelning. Räkna ut medelvärdet med `mean()` och spara varje medelvärde i en vektor. Visualisera hur medelvärdet förändras ju fler dragningar som görs med hjälp av `plot()` och `type="l"`, dvs antal dragningar på x-axeln och medelvärdet på y-axeln).

3.3. **Egenskaper hos väntevärden och varians.** I föreläsningarnas slides finns de (vanligaste) räknereglerna för väntevärden och varians. Vi ska nu undersöka dessa regler mer genom simulering. Med Monte Carlo metoder kan vi uppskatta det förväntade värdet med `mean()` och variansen med `var()` i R.

3.3.1. *Väntevärde och varians.* Vi ska utgå från följande två (oberoende) fördelningar

$$X \sim \text{Exp}(\theta = 10)$$
$$Y \sim \text{Poisson}(\lambda = 3)$$

(1) Besök Wikipedia för dessa två fördelningar för att ta reda på följande (teoretiska) storheter $E(X)$, $Var(X)$, $E(Y)$ och $Var(Y)$ och beräkna dessa teoretiska värden för $X$ och $Y$.
(2) Simulera 10 000 värden och beräkna $E(X)$, $Var(X)$, $E(Y)$ och $Var(Y)$ baserat på de utfall av fördelningarna du fått.
(3) Använd de teoretiska formlerna från föreläsningarnas slides (eller från Wikipedia) för att beräkna följande storheter (analytiskt):
- $E(3)$
- $E(3 \cdot X + 2)$
- $E(X + Y)$
- $E(X \cdot Y)$ (observera att $X$ och $Y$ är oberoende)
- $E(3 \cdot X + 2 \cdot Y - 3)$
- $Var(2 \cdot X - 5)$
- $Var(X + Y)$

(4) Upprepa nu samtliga beräkningar i (3) ovan men gör det genom simulering. **Tips!** Vill vi beräkna $E(2 \cdot Z - 5)$ där $Z \sim \mathcal{N}(\mu = 0, \sigma = 1)$ genom simulering gör vi på följande sätt i R. I exemplet nedan gör jag 10 000 dragningar.

```
> z <- rnorm(n = 10000, mean = 0, sd = 1)
> mean(2 * z - 5)

[1] -5.0298
```

Detta värde kan jämföras med det ”sanna” teoretiska värdet $E(2 \cdot Z - 5) = 2 \cdot E(Z) - 5 = -5$.

3.4. **Centrala gränsvärdessatsen.** Ovan såg vi att stora talens lag garanterade oss att medelvärdet av flera utfall av en slumpmässig variabel kommer konvergera mot det förväntade värdet då antalet dragningar går mot oändligheten. Centrala gränsvärdessatsen ger oss ett liknande resultat men gällande fördelningen för $\bar{X}$.

3.4.1. *Centrala gränsvärdessatsen.* Vi ska studera följande tre stokastiska variabler

$$\begin{aligned} X &\sim \text{Poisson}(\lambda = 5) \\ Y &\sim \text{Exp}(\theta = 1) \\ Z &\sim \text{Binom}(n = 10, p = 0.01) \end{aligned}$$

(1) Dra 1000 observationer från respektive fördelning och visualisera fördelningen med `hist()`.
(2) Skriv en loop/funktion i R som drar 10 värden från $X$ och $Y$ och beräknar medelvärdena, $\bar{X}$ och $\bar{Y}$, baserat på dessa 10 värden. Upprepa detta 1000 gånger och spara varje medelvärde i en vektor. Visualisera denna fördelning av medelvärden (dvs, vektoren) i ett histogram.
(3) Upprepa det du gjorde i uppgift (2) men dra nu 30, 100 och 1000 värden istället för 10 värden och visualisera resultatet i histogram. Vilken fördelning konvergerar medelvärdena mot? Vilken av de stokastiska variablerna $X, Y$ och $Z$ konvergerar snabbast mot denna fördelning? Varför?