

TBDA01: Laboration 1

Måns Magnusson, Mattias Villani

25 augusti 2017

Instruktioner

- Laborationen ska göras **två och två**
 - Labben ska vara en **PDF-rapport** med kod, analys och grafer. Ett tips är att använda R markdown. I rapporten ska följande ingå:
 - Båda studenternas namn och LiU-id.
 - Universitet och institution
 - Laborationsnummer
 - Deadlinen för laborationen framgår av [kurshemsidan](#).
 - Laborationsrapporten skickas in via LISAM.
 - För att få godkänt på laborationen krävs godkänt på **75%** av uppgifterna.
-

Innehåll

1	Introduktion till R	3
2	Laboration	4
2.1	Simulering och Monte Carlo-metoder	4
2.2	Stora talens lag	8
2.3	Egenskaper hos väntevärden och varians	8
2.4	Centrala gränsvärdessatsen	9

Kapitel 1

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](#).

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](#).

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](#).

Kapitel 2

Laboration

2.1 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. 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] 3.7605 2.2325 3.9832 3.9329 0.8325
```

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] 0.046677 0.039613 0.037212
```

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

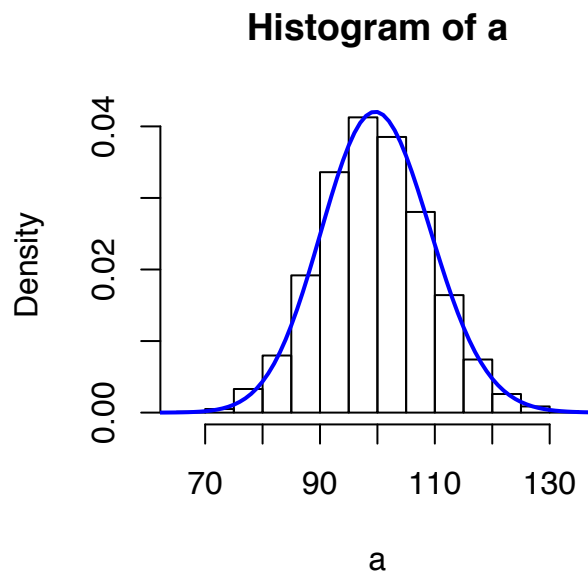
```
> z <- rbinom(n = 4, size = 1000, prob = 0.1)
> z

[1] 100 94 99 91
```

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

För att enkelt visualisera en fördelning kan vi använda funktionen `hist()` och fördelningens pdf 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)
```



Uppgift 1 Simulering av normalfördelning

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

- Visualisera fördelningarna i två histogram. Visualisera fördelningens pdf i samma graf.
- Beskriv skillnaden mellan de olika graferna.

Uppgift 2 Simulera och visualisera andra fördelningar

- Simulera och visualisera följande fördelningar med 10000 dragningar från varje fördelning samt fördelningens pmf/pdf.

Diskreta fördelningar

- $\mathbf{x}_1 \sim \text{Bernoulli}(p = 0.2)$
- $\mathbf{x}_2 \sim \text{Binomial}(n = 20, p = 0.1)$
- $\mathbf{x}_3 \sim \text{Binomial}(n = 20, p = 0.5)$
- $\mathbf{x}_4 \sim \text{Geometrisk}(p = 0.1)$
- $\mathbf{x}_5 \sim \text{Poisson}(\lambda = 10)$

Kontinuerliga fördelningar

- $\mathbf{y}_1 \sim \text{Likformig}(\min = 0, \max = 1)$
- $\mathbf{y}_2 \sim \text{Exp}(\theta = 3)$
- $\mathbf{y}_3 \sim \text{Gamma}(\alpha = 2, \beta = 1)$
- $\mathbf{y}_4 \sim \text{Student-t}(\nu = 3)$
- $\mathbf{y}_5 \sim \text{Beta}(\alpha = 0.1, \beta = 0.1)$
- $\mathbf{y}_6 \sim \text{Beta}(\alpha = 1, \beta = 1)$
- $\mathbf{y}_7 \sim \text{Beta}(\alpha = 10, \beta = 5)$

Tips! α heter **shape** och β 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. Det finns flera sådana relationer som är värdefulla att känna till.

Uppgift 3 Relation mellan fördelningar

Nedan finns tre stycken sannolikhetsfördelningar som konvergerar mot andra sannolikhetsfördelningar.

$$\mathbf{x}_1 \sim \text{Binomial}(n = 10000, p = 0.001)$$

$$\mathbf{x}_2 \sim \text{Student-t}(\nu = 10000)$$

- a) Simulera 1000 värden från respektive fördelning och visualisera fördelningen i ett histogram tillsammans med fördelningens pmf/pdf.
- b) Ta reda på (ex. via Wikipedia) vilken annan fördelning som respektive fördelning konvergerar mot.
- c) Simulera dragningar från dessa fördelning och jämför fördelningarna med de resultat du fick i a).

Uppgift 4 Hiearkiska sannolikhetsfördelningar

Vi ska nu simulera en blandad sannolikhetsfördelning (mixture distribution). Börja att läsa in datamaterialet `faithful` i R med `data(faithful)`.

- a) Visualisera variabeln `waiting` i ett histogram.
- b) Simulera från följande modell och visualisera fördelningen i ett histogram. Simulera först från X och sedan från Y . Sätt $p = 0.3$, $\mu_1 = 15$, $\sigma_1 = 3$, $\mu_2 = 4$ och $\sigma_2 = 2$.

$$x_i \sim \text{Bernoulli}(p)$$

$$y_i \sim x_i \cdot N(\mu_1, \sigma_1) + (1 - x_i) \cdot N(\mu_2, \sigma_2)$$

- c) Välj nu värden på $\mu_1, \mu_2, \sigma_1, \sigma_2$ och p så du får en liknande fördelning som för datamaterialet i variabeln `waiting`. Visualisera denna fördelning i ett histogram.

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.

Uppgift 5 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 täthetsfunktionen med prefixet `d`.

I denna uppgift ska vi arbeta med två fördelningar

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

- a) Använd täthetsfunktionen i R för att beräkna $P(Y = 0)$. Simulera nu 10000 dragningar från Y och beräkna andelen dragningar där $y = 0$.

Beräkna nu på liknande sätt $P(Y = 10)$ både med täthetsfunktionen och genom simulering.

- b) Använd nu den kumulativa fördelningsfunktionen i R för att beräkna följande sannolikheter.

1. $P(X < 0)$
2. $P(-1 < X < 1)$
3. $P(1.96 < X)$
4. $P(0 < Y < 10)$
5. $P(Y = 0)$
6. $P(0 \leq Y \leq 10)$

- c) Beräkna samma sannolikheter som i b) 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.

Uppgift 6 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 $\text{Bin}(n = 337, p = 0.1)$.

Det alternativ du står inför är betydligt mycket osäkrare, du vet inte hur stor felfrekvensen kommer bli så du uppskattar att felfrekvensen 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 $\text{Bin}(n = 337, p = P)$.

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 $[0, 1]$.

- a) Vad är det förväntade antalet fel för de båda systemen?
- b) Vad är sannolikheten att det nuvarande systemet genererar färre fel än det nya systemet?
- c) Vad är sannolikheten att du kommer få fler än 50 fel för respektive system?

"Monte carlo"-metoder kan också användas för att lösa andra problem än sannolikhetsberäkningar.

Uppgift 7 Monte Carlo metoder för integrering Simulera två likformiga fördelningar ($U(0, 1)$) av storlek $n = 1000$ som du kallar x och y . Ta bort de positioner ur vektorerna x och y där $\sqrt{x^2 + y^2} > 1$.

a) Visualisera de punkter som är kvar i en scatterplot. **Tips!** `plot()`

Räkna antalet punkter du har kvar och kalla detta för z .

Gör följande beräkning:

$$4 \cdot \frac{z}{n}$$

Upprepa beräkningen men simulera nu 10 000, 100 000 och 1 000 000 likformiga dragningar från $X \sim U(0, 1)$ och $Y \sim U(0, 1)$.

b) Vilken konstant har vi försökt approximera och hur väl tycker du vi lyckas?

c) Vad händer med approximationen när vi ökar antalet dragningar?

d) Beräkna $\int_0^2 x^2 dx$ analytiskt (eller med funktionen `integrate()` i R).

e) Simulera nu 100 000 värden från $X \sim U(0, 2)$ och $Y \sim U(0, 4)$ och beräkna samma integral med en Monte Carlo approximation.

2.2 Stora talens lag

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

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

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

Uppgift 8 Stora talens lag Vi ska studera följande två stokastiska variabler

$$X \sim \text{Bin}(n = 10, p = 0.2)$$

$$Y \sim \text{Gamma}(\alpha = 2, \beta = 2)$$

a) Använd wikipedia för att beräkna de (teoretiska) förväntade värdena ($E(X)$ och $E(Y)$) för dessa stokastiska variabler.

b) Dra 10, 20, ..., 90, 100, 200, ..., 900, 1000, 2000, ..., 9000 och 10000 antal värden från respektive fördelning, räkna ut medelvärde med `mean()` och spara varje medelvärde i en vektor. Visualisera hur medelvärde förändras ju fler dragningar som görs med hjälp av `plot()` och `type='l'` (d.v.s antal dragningar på x-axeln och medelvärde på y-axeln).

2.3 Egenskaper hos väntevärden och varians

I Mattias slides (slide 17) från föreläsning 2 finns de (vanligaste) räkneregler för väntevärden och varians. Vi ska nu undersöka dessa delar mer genom simulering. Baserat på Mattias slides kan vi beräkna det teoretiska förväntade värdet och den teoretiska variansen. Med Monte Carlo-metoder kan vi uppskatta det förväntade värdet med `mean()` och variansen med `var()` i R.

Uppgift 9 Väntevärde och varians

Vi ska utgå från följande två (oberoende) fördelningar

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

- a) Besök wikipediasidorna 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.
- b) Simulera 10 000 värden och beräkna $E(X)$, $Var(X)$, $E(Y)$ och $Var(Y)$ baserat på de utfall av fördelningarna du fått.
- c) Använd de teoretiska formlerna från Mattias material (eller från Wikipedia) för att beräkna följande storheter (analytiskt):
1. $E(3)$
 2. $E(3 \cdot X + 2)$
 3. $E(X + Y)$
 4. $E(X \cdot Y)$ (observera att X och Y är oberoende)
 5. $E(3 \cdot X + 2 \cdot Y - 3)$
 6. $Var(2 \cdot X - 5)$
 7. $Var(X + Y)$

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 10000 dragningar.

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

[1] -4.9897
```

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

- d) Upprepa nu samtliga beräkningar i c) ovan men gör det genom simulering.

2.4 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} .

Uppgift 10 Centrala gränsvärdessatsen

Vi ska studera följande tre stokastiska variabler

$$X \sim \text{Poisson}(\lambda = 5)$$

$$Y \sim \text{Exp}(\theta = 1)$$

$$Z \sim \text{Binom}(n = 10, p = 0.01)$$

- a) Dra 1000 observationer från respektive fördelning och visualisera fördelningen med `hist()`.
- b) 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 medelvärdena i en vektor. Visualisera denna fördelning av medelvärden i ett histogram.
- c) Upprepa det du gjorde i uppgift a) men dra nu 30, 100 och 1000 värden istället för 10 värden och visualisera resultatet. Vilken fördelning konvergerar medelvärdena mot? Vilken av de stokastiska variablerna X , Y och Z konvergerar snabbast mot denna fördelning? Varför?