Laboration 2

Statistik och dataanalys III, VT24

Mona Sfaxi

I Introduktion

I den här datorlabben kommer vi först att ägna oss mycket åt flera fördelningar som redan finns förprogrammerade i R, hur man kan utnyttja dem för beräkningar såväl som hur man kan illustrera dem grafiskt. Därefter kommer vi fokusera på samplingfördelningar. Innehållet i de 2 första avsnitten kan ses lite som repetition från statistik 1 med ett stort fokus på grafiska visualiseringar. Efter det kommer vi även titta lite på inversa funktioner och avslutningsvis kommer vi ägna oss åt Inverse transform sampling.

- ☐ De avsnitt som rekommenderas att göras markeras med ** eller *, där ** kan bedömas som en högre prioritering. De övningar som rekommenderas markeras med en *. Slutligen kan Avsnitt 3 och 4 ses som överkurs och behöver inte göras för att få vidare förståelse kring inlämningsuppgiften eller tentamen men finns där för den som är intresserad.

II Paket

Den här datorlabben kräver inte något paket. Alternativt kan du ladda ner `ggplot2` som även är en del av `tidyverse` paketet, ifall du skulle vilja använda dig av `ggplot()` för dina grafer.

1. Fördelningar i R

Vi har använt oss en hel del av Normal-, Binomial- och Poissonfördelningen under första terminen i statistik. Även om dessa fördelningar är väldigt härliga så finns många andra vanligt förekommande fördelningar förprogrammerade i R, både diskreta sådana och kontinuerliga som vi nu också kommer titta på.

1.1 d, p, q, r

Begynnelsebokstäverna d, p, q och r används för att hänvisa till olika användningsområden vad gäller alla fördelningar i R.

Används ett `d` framför funktionsnamnet avses täthetsfunktionen. Givet olika tal på `x` kommer funktionen att mata ut olika funktionsvärden `f(x)`. Exempelvis för normalfördelningen avser ett `d` framför `norm()` dess täthetsfunktion.

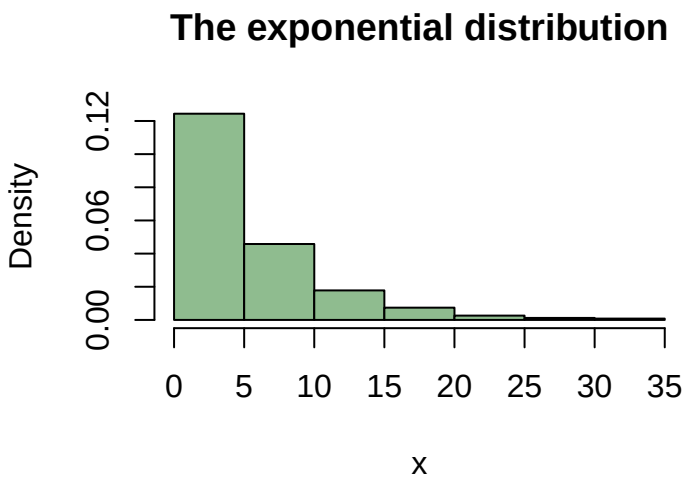
Används ett `p` framför funktionsnamnet avses istället fördelningsfunktionen (cumulative distribution function). Genom att skriva in ett visst kvantilvärde kommer funktionen att ge sannolikheten att man får ett värde mindre än eller lika med detta värde.

Ett `q` framför betecknar kvantilfunktionen. Givet en viss sannolikhet kommer funktionen producera fram kvantilvärdet på fördelningen.

Slutligen står `r` för random och används för att generera slumptal från fördelningen.

Innan vi går vidare med alla övningsuppgifter kan vi se hur alla dessa funktioner fungerar i ett exempel där vi använder oss av exponentialfördelningen. Låt $X \sim Exp(\beta = 5)$ så det teoretiska väntevärdet är 5. Vi kan börja med att simulera 1000 värden från denna fördelning med hjälp av `rexp(n = antal_simuleringar, rate)` och spara dessa värden under namnet xSim. Lägg märke till att vi här skriver `rate = 1/Beta`. Om vi hade skrivit `rate = Beta` hade vi fått den andra parametriseringen av exponentialfördelningen där väntevärdet $= 1/\beta = 1/5$. Därefter kan vi plotta våra observationer i ett histogram

```
Beta <- 5
xSim <- rexp(n = 1000, rate = 1/Beta)
hist(xSim, col = "darkseagreen", breaks = 10, xlab = "x", freq = FALSE,
      main = "The exponential distribution")
```



Vi använder argumentet `freq = FALSE` eftersom vi vill rita sannolikhetsfördelningen och inte titta på de absoluta frekvenserna.

Säg att vi nu vill undersöka grafiskt hur väl våra simulerade värden förhåller sig till den teoretiska fördelningen. Vi kan då rita en linje med den teoretiska fördelningen i samma graf. Hade vi inte skrivit `freq = FALSE` i histogrammet hade inte detta gått. För att rita den teoretiska fördelningen kan vi använda oss av funktionen `dexp(mina_x_värden, rate)`. Denna funktion kommer att producera funktionsvärden från den teoretiska fördelningen. Först måste vi dock skapa x-värden i sorterad ordning, annars kommer allt se väldigt konstigt ut. Vi vet att för exponentialfördelningen så måste $x > 0$. Vi har alltså ingen övre gräns för `x`. Men om vi ska rita detta måste vi själva bedöma vad som kan vara en rimlig gräns. I histogrammet ovan ser vi att de flesta värdena ligger mellan 0 och 40. Det kan vi använda. Så först skapar vi alltså en så kallad "grid" av x-värden som går mellan 0 och 40 som består av totalt 500 värden med hjälp av funktionen `seq(min, max, antal_obs)`

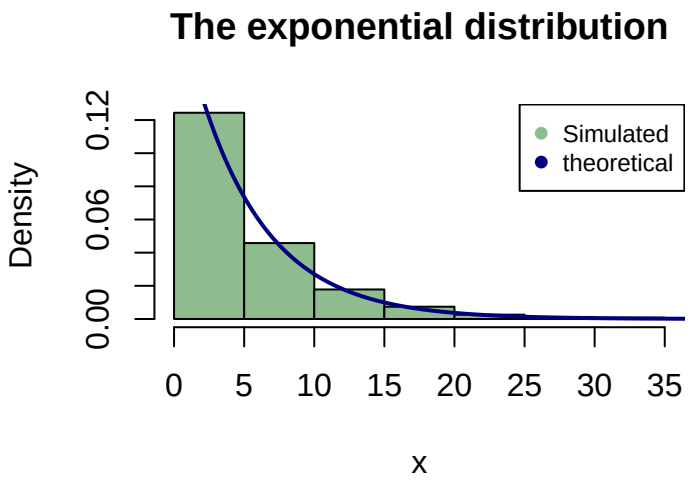
```
xGrid <- seq(0, 40, length = 500)
```

Nu kan vi använda dessa x-värden för att få fram våra funktionsvärden $f(x)$ i funktionen `dexp()`. Låt oss ge det ett passande namn så som `fx`

```
fx <- dexp(x = xGrid, rate = 1/Beta)
```

Därefter kan vi plotta våra simulerade värden tillsammans med våra teoretiska värden med hjälp av funktinen `lines(x, y)`. Vi börjar med att skapa histogrammet och lägger sedan till linjen och därefter en så kallad `legend` som innehåller etiketter för det vi har plottat:

```
hist(xSim, col = "darkseagreen", breaks = 10, xlab = "x", freq = FALSE,
     main = "The exponential distribution")
lines(x = xGrid, y = fx, col = "navy", lwd = 2)
legend("topright", legend = c("Simulated", "theoretical"),
       col = c("darkseagreen", "navy"), pch = c(19, 19), cex = 0.75)
```



Men hur ska vi göra ifall vi vill beräkna sannolikheter, som exempelvis $P(X \leq 4)$? Inga problem! Vi använder oss enkelt av funktionen `pexp(mitt_värde, rate)` som alltid beräknar sannolikheten att `X` är mindre än eller lika med ett visst värde.

```
pexp(4, rate = 1/5)
```

```
[1] 0.550671
```

Om vi istället vill beräkna $P(X > 5.5)$ då $X \sim Exp(\beta = 5)$ kan vi göra det genom att beräkna 1-komplementhändelsen, dvs: $P(X > 5.5) = 1 - P(X \leq 5.5)$

```
1 - pexp(5.5, rate = 1/5)
```

```
[1] 0.3328711
```

Men om vi nu istället vill veta vilket värde på x (dvs vilket värde på x-axeln) vi får om vi befinner oss i en viss percentil i fördelningen kan vi använda oss av funktionen `qexp(mitt_percentilvärde, rate)`. Låt säga att vi vill veta vilket värde på x-axeln vi skulle få ifall vi befann oss i den 90:e percentilen, vi gör då det med koden:

```
qexp(p = 0.9, rate = 1/5)
```

```
[1] 11.51293
```

Alltså skulle den 90:e percentilen motsvara ett värde på x-axeln som är lika med 11.52 för en exponentialfördelning med $\beta = 5$. Vi kan också kontrollera att det stämmer genom att lägga in värdet ovan i funktionen `pexp()`

```
pexp(11.51293, rate = 1/5)
```

```
[1] 0.9000001
```

Och ser att det finns ett samband mellan funktionerna `qexp()` och `pexp()`.

1.2 Normalfördelningen**

Låt oss börja lite kort med den allt annat än alldagliga fördelningen - Normalfördelningen. Som nämnt betecknas den som `norm` i R.

Uttrycket för normalfördelningens täthetsfunktion ges av

$$f(y) = \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \tag{1}$$

Exempelvis kan funktionen `dnorm(x, mean, sd)` användas för att få fram funktionsvärden för normalfördelingens täthetsfunktion. `dnorm()` fungerar både för ett värde på x såväl som för flera x-värden samtidigt:

```
dnorm(x = 2, mean = 0, sd = 1)
```

```
[1] 0.05399097
```

```
dnorm(x = c(2, -1.38), mean = 0, sd = 1)
```

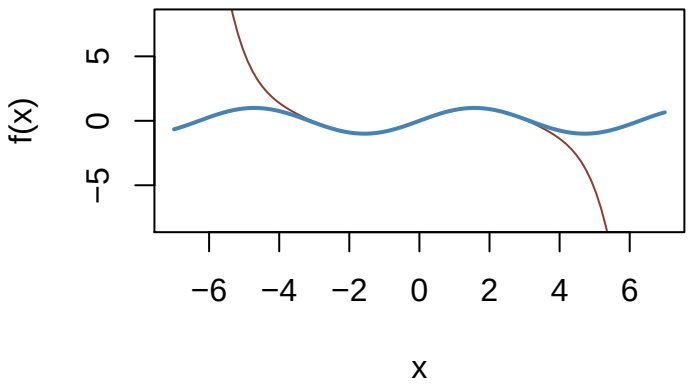
```
[1] 0.05399097 0.15394829
```

Detta blir särskillt användbart då man vill plotta sin fördelning.

> **Uppgift 1.1*** Låt $Y \sim N(\mu = 5, \sigma = 2.3)$. Använd funktionen `dnorm(x, mean, sd)` och låt Y vara lika med 4. Vad ger funktionen för värde? Är detta en sannolikhet?
>
> **Uppgift 1.2** Beräkna $P(Y \leq 6)$ för hand genom att standardisera och använda en tabell.
>
> **Uppgift 1.3*** Beräkna $P(Y \leq 4)$ genom att använda dig av en passande funktion i R.
>
> **Uppgift 1.4*** Vad är sannolikheten att Y ligger mellan 2.1 och 5.7? Använd R, men beräkna även gärna sannolikheten för hand för att bekräfta om du får tid över.
>
> **Uppgift 1.5*** Låt $Z \sim N(0, 1)$ använd R för att beräkna kvantilvärdet för den 5:e, 10:e och 97.5:e percentilen. Använd samma percentilvärden för Y och jämför nu dessa värden med dem för Z.
>
> **Uppgift 1.6*** Skriv en funktion i R som motsvarar uttrycket i Ekvation 1, som tar argumenten, `x`, `mu` och `sigma`. Låt `mu` ha default-argumentet 0 och `sigma` ha defaultargumentet 1.

Funktionen `curve(min_funktin, from, to)` är en annan rolig plot-funktion man kan använda för att rita funktioner som man har definierat själv. Fördelen är att man slipper skapa en massa x-värden som sedan används som argument i ens hemagjorda funktion. I exemplet nedan definierar vi två olika funktioner som vi sedan ritar i samma graf tillsammans. Detta görs genom att skriva `add=TRUE` i den nedre plotten som vi vill lägga till. Men lägg märke till att man kan behöva justera vilka värden på y-axeln som ska synas för att få en mer informativ graf. Utöver det använder `curve()` många av argumenten som används i funktionen `plot()`.

```
fun1 <- function(x) {
  x - x^3/factorial(3) + x^5/factorial(5) - x^7/factorial(7)
}
fun2 <- function(x) {
  sin(x)
}
curve(fun1, from = -7, to = 7, ylim = c(-8,8), col = "coral4", ylab = "f(x)")
curve(fun2, add = TRUE, col = 'steelblue', lwd = 2)
```



Uppgift 1.7* Använd funktionen `curve(min_funktion, from, to)`, för att rita upp din normalfördelning som du nyss har definierat. Fundera själv över vilka värden argumenten `from` och `to` bör ha så att du fångar det viktigaste i din fördelning.

Uppgift 1.8* Använd funktionen `integrate(fx, lower, upper)`, där fx är din definierade funktion från uppgift 1.6, för att beräkna sannolikheten att $-1.3 \leq x \leq 1.1$.

1.3 Den geometriska fördelningen**

Den geometriska fördelningen är en diskret fördelning och modellerar antal försök tills det första lyckade. Det bygger på oberoende, identiska försök med två möjliga utfall. Men det finns två olika sätt att represesentera fördelningen på. Låt `p` beteckna sannolikheten för ett lyckat försök. I kursboken används definitionen: "Antal försök tills att man får det första lyckade" (därav smeknamnet första gången fördelningen). Sannolikhesfördelningen ges då av

$$P(y) = (1-p)^{y-1}p, \qquad y = 1, 2, 3, ... \tag{2}$$

och medelvärdet ges av $E(Y) = \frac{1}{p}$.

Men det är också väldigt vanligt att definiera fördelningen som "Antal misslyckade försök tills att man får det första lyckade". Sannolikhetsfördelningen ges då av

$$P(y) = (1-p)^{y}p, \qquad y = 0, 1, 2, 3, ... \tag{3}$$

där medelvärdet istället ges av $E(Y) = \frac{1-p}{p}$.

I R används den senare definitionen. Den andra definitionen kan ses som en förskjutning av den första med ett steg. Det är också enkelt att koda fram den första definitionen genom att använda sig av en loop.

På Athena finns ett dataset "Drivers_licence" som vi nu kommer titta på. Detta dataset innehåller 103 observationer på olika individer som registrerat sig hos trafikverket för att ta ett B körkort och består av antal uppkörningar som de gjort under en 5 års period. Ladda ner datasetet och spara den i din arbetsmapp, helst i samma mapp som din Quarto fil. Tips: För att ladda datasetet i R, använd koden

```
mitt_data <- read.delim("Namn på filen.txt", header = TRUE, sep = " ")
```

där `mitt_data` är något lämpligt namn du kommit på själv.

Anta för enkelhetens skull att varje försök hos en enskild individ är oberoende av tidigare försök. Vi kan också anta oberoende mellan individer.

> **Uppgift 1.9*** Modellera antal uppkörningar det krävs att få godkänt med en geometrisk fördelning. Beräkna först den empiriska sannolikheten från datasetet att lyckas med uppkörningen i ett enskilt försök. Tips: Tänk på att
>
> $$E(Y) = \frac{1}{p}$$
>
> Så lös ut parametern `p`.
>
> **Uppgift 1.10*** Använd din skattning av `p` från ovan för att beräkna den teoretiska sannolikhetsfördelningen för y = 1, 2,..., 10 med hjälp av en for-loop (här kan vi ignorera värden större än 10 eftersom de inte är så sannolika). Glöm inte att spara sannolikheterna i en vektor. Tips: du kommer behöva anväda dig av uttrycket i Ekvation 2 ovan.
>
> **Uppgift 1.11*** Beräkna `fördelningsfunktionen` för $y = 1, 2, ..., 10$ genom att beräkna den kumulativa summan av sannolikheterna från uppgift 2.10. Här kan man med fördel använda funktionen `cumsum(mina_sannolikheter)`
>
> **Uppgift 1.12*** Illustrera den empiriska fördelningen i en passande graf. För att göra detta behöver du först räkna den relativa frekvensen för varje utfall. Det kan enkelt göras med hjälp av koden `prop.table(table(min_variabel))`. Tips: för att plotta, använd pseudo-koden:

```
plot(mina_relativa_frekvenser)
```

Uppgift 1.13* Lägg till en linjeplot med de teoretiska sannolikheterna genom att använda dig av funktionen `lines(y, fy)` under din plot. Kom ihåg att du måste köra `lines()` funktionen samtidigt som du kör kommandot för `plot()` i Quarto, annars kommer R inte att visa något. Verkar datasetet vara bra anpassat till din modell?

Uppgift 1.14* Skapa en till plot, denna gång över den teoretiska fördelningsfunktionen Använd gärna kommandot `plot(mina_kumulativa_frekvenser, type = "s")`. Varför ser den ut så? Är det rimligt?

1.4 Poissonfördelningen*

Då man har räknedata, som till exempel antal klick på en annons per dag, kan man ofta använda sig av Poissonfördelningen. Den ges av.

$$P(Y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

Uppgift 1.15* Om $Y \sim Pois(\lambda = 4)$ vad är sannolikheten att $Y \geq 5$?. Vad är sannolikheten att $2 \leq Y \leq 4$? Använd R för din beräkning men bekräfta gärna resultatet för hand.

Uppgift 1.16 Vilket värde på Y motsvaras av den 17:e, 23:e och 30:e percentilen från denna fördelning? Vad kan förklara resultatet?

Uppgift 1.17* Låt $U \sim Pois(3)$. Simulera 10 000 observationer från denna fördelning och rita upp den i en passande graf. Här är det inte så passande att använda sig av ett histogram eftersom `A` är diskret och består av få värden (och flera värden kan då komma att slås ihop i större klasser i histogrammet vilket skulle vara sorgligt). Ett tips är därför att först skapa en tabell med proportioner för varje tal.

Uppgift 1.18* Låt också $V \sim Pois(3)$ och $W \sim Pois(3)$. Simulera 10 000 observationer vardera och rita upp dem båda i egna grafer precis som ovan.

Uppgift 1.19* Låt nu $UVW = U+V+W$. Beräkna medelvärdet och variansen av den nya variabeln och illustrera den i en passande graf. Vad följer din nya variabel för fördelning?

Företaget Kepler tillverkar ledlampor. De tillverkar miljontals sådana varje år. Tillverkningen av lamporna kan antas vara oberoende av varandra. Varje dag tas ett stickprov om 10 stycken där man testar ifall varje lampa fungerar som de ska. Sannolikheten att en lampa är trasig är 0.005.

Uppgift 1.20* Hur stor är sannolikheten att högst 1 lampa är trasig? Använd dig av binomialfördelningens fördelningsfunktion som ges av `pbinom(q = x, size = n, prob = p)` för att beräkna det här.

En inspektör hos företaget upptäckte att en maskin i tillverkningsprocessen inte verkade fungera som den skulle under 12 timmar. Eftersom företaget skulle skicka en stor leverans till en viktig kund ville de veta ifall det var något fel på lamporna. Denna gång tog de ett urval på 1000 stycken lampor och beslutade sig för att skrota samtliga lampor som tillverkats under dagen ifall fler än 10 stycken var trasiga.

Poissonfördelningen användas ofta som approximation av Binomialfördelningen då `n` är högt och `p` väldigt litet.

Uppgift 1.21* Beräkna sannolikheten att fler än 10 lampor är trasiga med hjälp av en Poisson-approximation. Tips: Räkna först ut vad λ bör vara utifrån din approximation.

Uppgift 1.22 - (Extrauppgift) Rita upp dina två fördelningar i en graf tillsammans. Tips: Beräkna först vad λ är och skapa därefter x-värden som går från 0 till 1000. Använd sedan dessa värden i dina två frekvensfunktioner där `pmf1 = dbinom(x, n, p)` och `pmf2 = dpois(x, lambda)`. Använd sedan funktionerna `plot(x, pmf)` och `lines(x, pmf)` för att illustrera båda fördelningar i samma graf. Justera gärna så att x-axeln endast visar tal från 0 till 30 med hjälp av funktionen `xlim = c(minsta_vädet, högsta_värdet)`. Är approximationen bra?

Ofta används Poissonfördelningen vid så kallade "räknedata". Men Poissonfördelningen kan vara begränsad i och med att medelvärdet bör vara lik variansen. I riktiga data däremot brukar variansen oftast vara större än medelvärdet. Den negativa binomialfördelningen används ibland för att moddellera räknedata då variansen tros vara större än medelvärdet.

1.5 Negativ Binomialfördelning*

Den geometriska fördelningen kan ses som ett specialfall av den negativa binomialfördelningen. Den är också diskret och bygger på oberoende, identiska försök men liksom för den geometriska fördelningen finns också flera olika möjliga definitioner. Fördelningen har två parametrar - p och r. Låt p vara sannolikheten att lyckas i ett enskilt försök. I boken används definitionen "Antal försök tills att vi får det r:te lyckade" och sannolikhetsfördelningen ges utav

$$P(y) = \binom{y-1}{r-1} p^r (1-p)^{y-r}, \qquad y = r, r+1, r+2, r+3, ... \tag{4}$$

I R används istället definitionen "Antal misslyckade försök tills att vi får det r:te lyckade" så se upp innan du använder denna fördelning när du programmerar!

Vi ska nu använda R's definition av sannolikhetsfördelningen som ges utav

$$P(y) = \frac{(y+r-1)!}{(r-1)!y!}p^r(1-p)^y, \qquad r > 0, \quad y = 0, 1, 2, ... \tag{5}$$

I R används funktionen `dnbinom(x, size, prob)` där `x` representerar ett realiserat värde på din slumpvariabel (eller med andra ord; ett faktiskt värde på y i ekvationen ovan), `size` motsvarar `r` och är alltså antal lyckade försök och `prob` är sannolikheten för ett enskilt lyckat försök.

Uppgift 1.23* Antag att $Y \sim NegBin(r = 3, p = 0.4)$. Använd dig av funktionen `dnbinom()` och låt Y vara lika med 1. Är detta en sannolikhet? Jämför ditt svar med motsvarande fråga för normalfördelningen. Varför blir det så?

Uppgift 1.24 Använd pnbinom för att beräkna $P(1 \leq Y \leq 3)$ givet att $Y \sim NegBin(r = 3, p = 0.4)$.

Uppgift 1.25* Simulera 10 000 observationer från en negativ binomialfördelning med r = 20 och p = 0.35 (använd definitionen i Ekvation 5). Illustrera din nya variabel i en passande graf tillsammans med en linje över den teoretiska fördelningen. Beskriv utseendet på fördelningen.

Uppgift 1.26* Tycker du att det är konstigt att exempelvis beräkna $P(Y \leq 2)$ ifall $r = 4$ genom att använda definitionen i Ekvation 4? Vad innebär denna definition? Går det att beräkna $P(Y \leq 2)$ då $r = 4$ ifall man istället använder definitionen i Ekvation 5? Här behöver du inte göra några beräkningar ifall du inte vill, men det kanske kan hjälpa i resonemanget.

2. Samplingfördelningar

En samplingfördelning är kort sagt en sannolikhetsfördelning för en statistika. Den mest bekanta statistikan är förmodligen urvalsmedelvärdet, men även urvalsvariansen såväl som maximum eller minimum värdet är vanligt förekommande.

2.1 Samplingfördelningen för Urvalsmedelvärdet**

En slutsats man kan dra av den centrala gränsvärdessatsen (CGS) är att fördelningen för urvalsmedelvärdet $E(\overline{Y})$ är normalfördelad med parametrarna μ och $\frac{\sigma^2}{n}$ oavsett vilken fördelning Y kommer ifrån, så länge som Y har ett medelvärde och en ändlig varians.

Vi ska nu titta på samplingfördelningen för medelvärdet hos en gammafördelning. Gammafördelningen ges av

$$\frac{1}{\Gamma(\alpha)\beta^\alpha}y^{\alpha-1}e^{-\frac{y}{\beta}}$$

Uppgift 2.1* Börja med att simulera 10 000 observationer från en gammafördelning med parametrarna $\alpha = 5$, $\beta = 7$ och rita upp fördelningen i ett histogram. Tips: använd funktionen `rgamma(n, shape, scale)`. där n står för antal simuleringar, `shape` står för α och `scale` står för β. Om du använder `rate` istället för `scale` så kommer du använda en annan definition av gammafördelningen än den som används i kursen.

Uppgift 2.2* Skapa sedan en tom vektor vid namn `samp5` som du fyller med 1000 nollor. Använd gärna funktionen `rep(0, n = Antal_simuleringar)` såsom i labb 1 för att göra detta. Skapa sedan en loop. I loopen ska du dra ett urval av storlek 5 ur ditt simulerade dataset ovan. Använd gärna funktionen
`sample(x = min_variabel, urvalsstorlek, replace = TRUE)` för att dra ett slumpmässigt urval. Beräkna sedan medelvärdet av dina 5 observationer (du befinner dig fortfarande i loopen) och spara sedan detta urvalsmedelvärde i variabeln `samp5`.

Uppgift 2.3* Beräkna medelvärdet och variansen av `samp5` efter att du har loopat klart och rita ett histogram över din nya fördelning. Använd dig av ett lämpligt antal "breaks" i ditt histogram.

Uppgift 2.4* Upprepa samma procedur som ovan men dra denna gång ett större urval i din loop på 20 respektive 50. Döp dessa variabler till `samp20` och `samp50`. Jämför utseendet på fördelningarna med varandra såväl som medelvärdet och variansen. Vad bör den teoretiska variansen vara enligt CGS?

Uppgift 2.5* Beräkna även bredden på intervallet mellan den 97.5:e och 2.5:e percentilen för alla tre urval. Tips: använd funktionen
`quantile(min_variabel, min_percentil)`. Vad händer med fördelningen för medelvärdet då urvalsstorleken `n` går mot oändligheten?

2.2 Samplingfördelning för urvalsvariansen*

Om X_i är en normalfördelad variabel med $\mu = 10$ och $\sigma^2 = 9$ så är $Z_i = \frac{X_i - \mu}{\sigma} \sim N(0, 1)$ Dvs Z_i är standard normal. Men vad händer om man tar kvadraten av en variabel som är standard normal och sedan summerar den tillsammans med n stycken andra variaber som också är standard normala? Spännande nog så får vi en $\chi^2(n)$ fördelad variabel, dvs

$$\sum_{j=1}^{n} Z_j^2 \sim \chi^2(n)$$

Vi ska nu visa att så är fallet med hjälp av simuleringar.

Uppgift 2.6* Börja med att skapa en matris av dimension 1000×5 (dvs 1000 rader och 5 kolumner) som du fyller med 1000×5 observationer som alla är normalfördelade med $\mu = 10$, och $\sigma^2 = 9$. Kalla matrisen för `X`. Det finns olika sätt man kan fylla matrisen på. Man kan använda sig av en loop och exempelvis fylla varje kolumn med 1000 observationer. Men ett mer effektivt sätt är att använda sig av koden

```
matrix(data = rnorm(Antal_observationer * Antal_kolumner, mu, sigma),
       nrow = Antal_observationer, ncol = antal_kolumner)
```

Uppgift 2.7 Skapa sedan en till matris `Z` genom att subtrahera det teoretiska medelvärdet 10 från varje observation i matrisen X och sedan dividera detta med den teoretiska standardavvikelsen 3. Undersök gärna fördelningen inom varje kolumn för Z.

Uppgift 2.8 Bilda en ny variabel `Chi2_5` som helt enkelt är radsumman av Z^2. Tips: använd funktionen `rowSums(min_matris)` för att beräkna radsumman.

Uppgift 2.9 Rita upp fördelningen för `Chi2_5` i en passande graf och lägg till den teoretiska linjen för en $\chi^2(5)$ fördelning. Tips: I R ges täthetsfunktionen för $\chi^2(n)$ fördelningen av funktionen `dchisq(x = min_variabel, df = n)`.

Ovan har vi utgått från att det teoretiska medelvärdet är känt och vi fick ett mycket trevligt resultat. Men om man inte känner till det teoretiska medelvärdet, vad gör man då? Inga problem då vi vet att $\sum_{j=1}^{n} \frac{(X_j - \overline{X})^2}{\sigma^2} \sim \chi^2(n-1)$ (dvs vi förlorar en frihetsgrad eftersom vi först måste skatta medelvärdet). Men kom ihåg att σ^2 är en konstant så vi kan istället skriva $\frac{1}{\sigma^2}\sum_{j=1}^{n}(X_j - \overline{X})^2 \sim \chi^2(n-1)$. Men lägg märke till att urvalsmedelvärdet $S^2 = \frac{\sum_{j=1}^{n}(X_j - \overline{X})^2}{n-1}$ så $(n-1)S^2 = \sum_{j=1}^{n}(X_j - \overline{X})^2$. Alltså får vi

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1) \tag{6}$$

Vi ska nu visa att detta stämmer med hjälp av simuleringar.

Uppgift 2.10* Börja med att beräkna urvalsvariansen `S^2` för varje rad i matrisen `X` som du skapade innan. Här kan man använda en for-loop men det går också smidigt att använda sig av funktionen `apply()` istället (se datorlabb 1). Skriv i så fall MARGIN = 1 och använd dig av funktionen `var()` för att beräkna urvalsvariansen. Bilda sedan en ny variabel `chi2_4` som helt enkelt är lika med uttrycket i Ekvation 6, dvs, produkten av (antal frihetsgrader-1) och urvalsvariansen dividerat med den teoretiska variansen.

Uppgift 2.11* Rita fördelningen i en passande graf tillsammans med den teoretiska linjen för en $\chi^2(4)$ fördelning. Stämmer medelvärdet från din fördelning med det teoretiska medelvärdet?

2.3 En kärlekshistoria mellan en standard normalfördelning och en χ^2-fördelning*

Vi ska nu titta vidare på vad som händer om man dividerar en standard normalfördelning med $\sqrt{\chi^2(n)/n}$. Låt $Z \sim N(0,1)$ och $W \sim \chi^2(n)$ Utifrån sannolikhetsteori bör detta ge oss en t-fördelning med n frihetsgrader.

Innan vi undersöker detta resultat så kommer vi först rita upp t-fördelningens täthetsfunktion. I R betecknas denna som `dt(mina_x_värden, df = mina_frihetsgrader)`. Skapa en så kallad `grid` av minst 100 x-värden som går från -4 till 4. Illustrera sedan t-fördelningen med 2, 10, 30 och 100 frihetsgrader i ett linjediagram. Lägg till ytterligare en linje i samma graf, denna gång över en standard normalfördelning för en sista touch. Glöm inte att särskilja på linjerna genom att använda olika färger. Lägg också gärna till en så kallad `legend()` för att ge dina linjer etiketter, så att man kan se vilka fördelningar alla linjer tillhör. Vad händer med fördelningarna då antal frihetsgrader ökar? Varför tror du att detta är betydelsefullt i praktiken?

Nu kan vi fortsätta med simuleringsexperimentet.

> **Uppgift 2.12*** Simulera 10 000 observationer från en standard normalfördelning och döp denna variabel till Z.
>
> **Uppgift 2.13*** Simulera sedan 10 000 observationer från en $\chi^2(5)$-fördelning och döp denna till W.
>
> **Uppgift 2.14*** Bilda en ny variabel `T5` genom att ta $\frac{Z}{\sqrt{W/5}}$.
>
> **Uppgift 2.15*** Rita upp din variabel `T5` i ett histgram tillsammans med den teoretiska linjen för t-fördelingen med 5 frihetsgrader. Ser det ut att stämma?

Låt oss uppehålla oss lite mer vid t-fördelningen som du säkert känner igen med formeln

$$\frac{\overline{X} - \mu}{S/\sqrt{n}} \sim t_{(n-1)}$$

På Athena finns ett dataset; `curl` som representerar vikten som lyfts bland manliga medlemmar i skivstångscurl på ett privat gym i en bostadsrättsförening. Vikterna antas vara normalfördelade. Förutom det vet man också att $\mu = 46.6$.

Du ska nu dra olika stora urval från `curl`, beräkna medelvärdet från varje enskilt urval och standardisera resultatet. Du ska alltså beräkna

$$\frac{\overline{X} - \mu}{S/\sqrt{n}}$$

för olika storlekar på `n` och sedan illustrera dessa fördelningar grafiskt.

> **Uppgift 2.16** Ladda ner dataseten med hjälp av koden

```
min_variabel <- readRDS("mitt_dataset.rds")
```

Alternativt, ifall du använder en textfil istället med koden:

```
min_variabel <- read.delim("mitt_dataset.txt")
```

Uppgift 2.17 Skapa två tomma vektorer och kalla dem för något passande som exempelvis `xbar5` och `sd5` och fyll vardera med minst 1000 nollor. Skapa sedan en for-loop och använd funktionen `sample(min_variabel, size = urvalsstorlek, replace = TRUE)` innuti loopen för att i varje iteration dra 5 stycken observationer från datasetet. Beräkna sedan medelvärdet och standardavvikelsen (fortfarande innuti loopen) från dina 5 observationer och spara dem i `xbar5` och `sd5`.

Uppgift 2.18 Bilda sedan en ny variabel `t5_data`, utanför din loop, genom att subtrahera det sanna $\mu = 46.6$ från `xbar5` och sedan dividera denna differens med `sd5` dividerat på $\sqrt{5}$.

Uppgift 2.19 illustrera därefter fördelningen för `t5_data` grafiskt tillsammans med en linje med den teoretiska t-fördelningen och kommentera eventuella likheter och skillnader (Observera att den teoretiska fördelningen inte är $\sim t(n)$). Beräkna även det empiriska medelvärdet av `t5_data`, är det nära det teoretiska?

Uppgift 2.20 Gör nu samma sak som ovan fast med ett större urval på 30. Jämför även ditt resultat med resultatet du fick ovan.

3. Icke-monotona och Inversa funktioner

Ibland träffar vi på funktioner eller sannolikhetsfördelningar som ser väldigt konstiga ut, som är definerade på olika sätt i olika stycken av funktionen och som inte är monotona. Men hur kan man rita en sådan funktion? Låt $f(y)$ vara en sådan funktion. Man kan då multiplicera definitionen för $f(y)$ med dess definitionsvärden likt principen nedan

```
fun <- function(y){
  f(y)*(y <= något definitionsvärde) + f(y)*(y > något definitionsvärde)
}
```

Ett exempel på hur denna princip kan fungera i praktiken ges för den konstiga funktionen $f(y)$ som är icke-monoton:

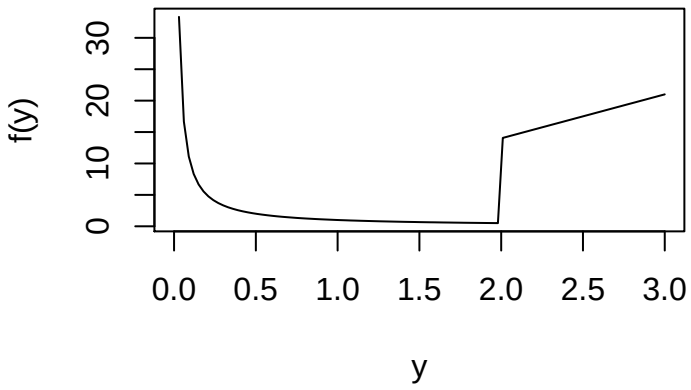
$$f(y) = \left\{ \begin{array}{cc} \frac{1}{y}, & 0 < y \leq 2 \\ 7y, & y > 2 \end{array} \right\}$$

För att kunna rita den här funktionen kan vi göra som i koden nedan. Först multiplicerar vi definitionsmängden med dess respektive så kallade "styckfunktion" och sedan plussar vi ihop

de två styckfunktionerna. Därefter kan vi rita allt som en sammanhängande funktion i en graf.

```
weird_fun <- function(y){
  1/y*(y <= 2) + (7*y)*(y>2)
}

curve(weird_fun, from = 0, to = 3, ylab = "f(y)", xlab = "y")
```



Lägg märke till att den här funktionen inte är en sannolikhetsfördelning.

Transformationsmetoden (Finns ej i kursmaterialet längre men går att läsa om i kursboken under avsnitt 6.4) är ett bra verktyg för att finna täthetsfunktionen av en stokastisk variabel som är en funktion av en annan stokastisk variabel. Det finns dock några krav som måste uppfyllas innan man kan använda sig av metoden. Först och främst måste funktionen vara deriverbar och en-entydig (monoton) för definitionsområdet. Är den inte monoton för hela definitionsområdet så kan man dela upp den så att den blir monoton för olika delar av hela definitionsområdet. Sedan måste det även vara möjligt att få fram den inversa funktionen.

Vi ska nu titta lite på några inversa funktioner och illustrera dem grafikt. Låt u vara en funktion av y, dvs $u = f(Y)$ i dessa tre exempel nedan.

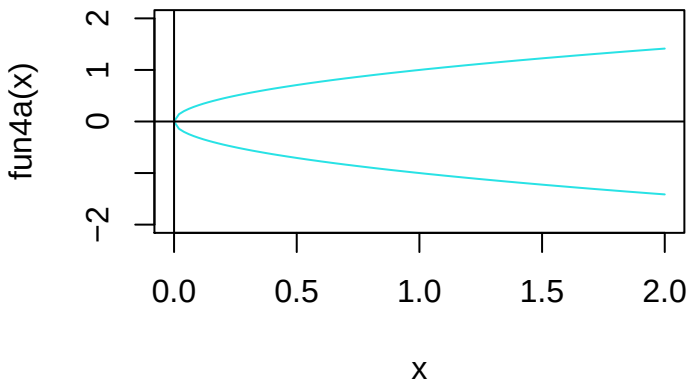
1. $u = y^2, \quad -2 \leq y \leq 2$
2. $u = e^{-\frac{y}{3}}, \quad y > 0$
3. $u = |2y|, \quad -1 \leq y \leq 1$

Uppgift 3.1 Illustrera y och u i de tre exemplena ovan i samma graf genom att först skapa funktioner för dem i R. Låt y vara på x-axeln och u på y-axeln. Finns det ett monotont förhållande mellan u och y i samtliga exempel? Om inte så får du gärna dela upp funktionerna med hjälp av gränserna för y.

Uppgift 3.2 Beräkna (för hand) värden som u är definierad för (Tips lägg in definitionsgränserna för y i f(y)). Beräkna därefter inversen av u i respektive funktion, genom att lösa ut y och illustrera dessa funktioner i en annan graf. För funktion 1 där $u = y^2, \ -2 \leq y \leq 2$ måste man skapa två separata funktioner för dess invers likt nedan:

```
fun4a <- function(y) -sqrt(abs(y))*(y>=0)
fun4b <- function(y) sqrt(abs(y))*(y>=0)

curve(fun4a, from = 0, to = 2, col = 5, ylim = c(-2,2))
curve(fun4b, add = TRUE, col = 5)
abline(h = 0, v = 0)
```



Fundera gärna noga på vilka värden på U som inversen $h^{-1}(u)$ är definierad för hos de andra funktionerna när du definierar dem i R och ha endast med dessa värden på "u". Egentligen kommer det gå att rita dessa funktioner ändå, även fast du inte har multiplicerat funktionerna med deras definitionsvärden. Men i nästa avsnitt när vi simulerar från en Laplacefördelning kommer du se varför det kan vara betydelsefullt att göra på det här sättet.

4. Inversa transformationsmetoden

Om en variabel Y har täthetsfunktionen f(y) och fördelningsfunktion F(y) så kan man visa att F(y) är likformigt fördelad mellan 0 och 1. Det är alltså lika mycket sannolikhetsmassa mellan olika percentiler. Exempelvis är det lika mycket sannolikhetsmassa mellan den 10:e och 20:e percentilen som det är mellan den 60:e och 70:e percentilen. Det kanske inte låter jätte roligt men i praktiken så innebär detta att ifall vi känner till en variabels fördelningsfunktion och kan beräkna inversen av dess fördelningsfunktion (dvs lösa ut Y ur U = F(y)) analytiskt så kan vi också simulera observationer från dess täthetsfunktion genom att använda oss av slumpmässigt likformigt fördelade observationer, oavsett hur konstig F(y) ser ut att vara! Låt oss testa!

4.1 Laplacefördelningen (Avancerat)

Under senare delen av Statistik 1 ägnade ni er åt regularisering där ni använde funktionen `glmnet()` för att utföra `Ridge` och `Lasso` regression. `Lasso regression` utförs med hjälp av en `Laplace` prior (mer om vad en prior är kommer under datorlab 4 så håll ut!). Laplacefördelningen är något lik normalfördelningen i och med att den är symmetrisk och har definitionsområde $-\infty \leq y \leq \infty$. Men till skilland från normalfördelningen så är den spetsig i mitten och har fetare svansar. Täthetsfunktionen ges av

$$f(y) = \frac{1}{2b} e^{\left(-\frac{|y-\mu|}{b}\right)}$$

Där μ är en lokaliseringsparameter, och anger vilket värde sannolikhetsfunktionen är symmetrisk kring, och b är en skalparameter. Ju större b, desto tjockare svansar. Den här fördelningen finns inte förprogrammerad i R, men lyckligtvis så kan vi simulera den med hjälp av den inversa transformationsmetoden.

> **Uppgift 4.1** Skapa först en funktion för sannolikhetsfördelningen (dvs täthetsfunktionen) med default-värdena $\mu = 0$ och $b = 1$ och rita sedan den i en graf för $-7 \leq y \leq 7$ så att du kan bekanta dig med den. Du ska nämnligen simulera från denna fördelning i nästa steg.

Fördelningsfunktionen $F(y)$ ges av

$$\begin{aligned} \frac{1}{2} e^{\left(\frac{y-\mu}{b}\right)}, &\quad y \leq \mu \\ 1 - \frac{1}{2} e^{\left(-\frac{y-\mu}{b}\right)}, &\quad y > \mu \end{aligned}$$

> **Uppgift 4.2** Låt $U = F(y)$, dvs fördelningsfunktionen för y som ges ovan. Finn inversen för U genom att lösa ut y (som du ser ovan så kommer uttrycket att bero på hur y förhåller sig till μ). Skapa sedan en funktion i R med detta uttryck och kalla funktionen för `u_inverse`. Funktionen ska ha argumenten `u, mu och b`.

Uppgift 4.3 Simulera därefter 10 000 observationer från en likformig fördelning (0, 1) med hjälp av funktionen `runif(n, min, max)`och spara dessa värden i en vektor med namnet `u`.

Uppgift 4.4 Simulera från `u_inverse(u, mu = 0, b = 1)` funktionen genom att använda vektorn `u` som argument. Spara dina observationer och rita upp dem i ett histogram tillsammans med den teoretiska täthetsfunktionen för Laplacefördelningen. Lek gärna runt genom att ändra på värdena för `mu` och `b`.

Uppgift 4.5 Beräkna (den empiriska) sannolikheten att $X > 2$.

Uppgift 4.6 Använd funktionen `integrate(fx, lower, upper)` för att jämföra den teoretiska sannolikheten ovan, där fx är den teoretiska täthetsfunktionen som du skapade innan du började simulera. Observera att du inte kan skriva `integrate(fx(x), lower, upper)`.

Uppgift 4.7 Vad tror du nackdelen är med att använda sig av den inversa transformationsmetoden för att simulera från en sannolikhetsfördelning?

5. Sammanfattning

I den här datorlabben har vi bekantat oss med olika fördelningar och hur man kan utnyttja R's inbyggda funktioner för att göra olika beräkningar. Vi har även tittat på olika samplingfördelningar och använt oss av en hel del av simuleringar i avsnitt 2. I det tredje avsnittet har vi beräknat inverser av olika funktioner och visualiserat dem. Avslutningsvis har vi använt oss av inverse transform sampling för att simulera från fördelningar som kanske inte är lika vanligt förekommande.