

## Tentamen i Programmering i R, 7.5 hp

---

Skrivtid: 8-12  
Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2" av Matt Baggot, referenskort för ggplot2 och RStudio tillgängligt elektroniskt.  
Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt, 16 poäng ger Väl godkänt.

Tänk på följande:

Skriv dina lösningar i **fullständig och läsbar kod**.

Lösningen skrivs i en körbar textfil med namnet **Main.R**.

Se filen **DocStudent.pdf** för hur tentan ska lämnas in.

**Kommentera direkt i Main.R** filen när något behöver förklaras eller diskuteras.

Eventuella grafer som skapas under tentans gång behöver **INTE** skickas in för rättning, det räcker med att **skicka in den kod som producerar figurerna**.

---

**OBS: Glöm inte att spara din fil ofta! Om R krashar kan kod förloras.**

### 1. Datastrukturer och beräkningar (4p)

- (a) Beräkna  $y = \sin\left(\frac{x^3}{\log_{10}(x)}\right)$  där  $x = 0.5$  och  $\log_{10}$  betyder logaritmen med bas 10. Spara resultatet i variabeln **y**. **1p**
- (b) Läs in det interna datamaterialet **trees** och räkna ut det variansen för variabeln **Height**, men bara för de observationer där variabeln **Girth** är större än eller lika med 11. Spara värdet i variabeln **my\_var**. **1p**
- (c) Skapa nu en matris med 3 rader och 10 kolumner. Matrisen ska vara fylld med slumpstal från en Poissionfördelning med parameter  $\lambda = 2$ . Du får själv bestämma vilken seed som används. Spara matrisen i **my\_mat**. **1p**
- (d) Nu ska en lista skapas som innehåller variablerna som skapades i a)-c). Listan ska fyllas med **y**, **my\_var** och **my\_mat** (i given ordning). Lis-telemetens namn ska vara samma som variabelnamnen. Spara listan i **my\_list**. **1p**

2. Kontrollstrukturer (4p)

- (a) Skapa en tom matris  $A$  av storlek  $3 \times 4$  och fyll den sedan med elementen  $a_{ij} = j \cdot \cos(\pi \cdot i) / i$  där  $i$  är radnumret och  $j$  är kolumnnumret. Du ska använda en nästlad **for**-loop. **2p**
- (b) Skapa en **while**-loop som skriver ut de 10 första heltalen som är jämt delbara med 13, **print()** ska användas. **2p**

3. Strängar, datum och funktioner (4p)

- (a) Albert Einstein föddes 14 mars 1879 och avled 18 april 1955. Florence Nightingale föddes 12 maj 1820 och avled 13 augusti 1910. Använd **lubridate** för att svara på frågorna nedan. **2p**
  - i. Hur många hela veckor levde Florence innan Albert föddes?
  - ii. Hur många hela månader levde Florence och Albert samtidigt?
  - iii. Hur många dagar levde Florence och Albert sammanlagt? (inkludera dagen de avled)
  - iv. Vilken veckodag avled Florence på?
- (b) Nu ska en funktion skapas som kan derivata ett polynom analytiskt. Om  $y = f(x) = a \cdot x^n$  då är derivatan  $\frac{dy}{dx} = a \cdot n \cdot x^{n-1}$ , där  $a$  är en reel konstant och  $n$  är ett heltal. Skapa nu funktionen **ploy\_derivative(f=)**, som ska ta en textsträng som argument. Ex **f="2x^3"**. Funktionen ska då returnera uttrycket för derivatan som en textsträng. Utgå ifrån att variabeln alltid heter **x**.  $a$  antas vara positivt heltal och  $n$  är ett heltal mellan 1 och 9.  $a$  antas kunna bestå av godtyckligt antal tecken. Funktionen ska returnera en text-sträng på följande enligt testfallen nedan. **2p**

```
ploy_derivative(f = "3x^1")

[1] "dy/dx=3x^0"

ploy_derivative(f = "2031x^2")

[1] "dy/dx=4062x^1"

ploy_derivative(f = "25x^3")

[1] "dy/dx=75x^2"

ploy_derivative(f = "8x^9")

[1] "dy/dx=72x^8"
```

#### 4. Statistik och funktioner (4p)

- (a) Nu ska en funktion skapas som kan beräkna dubbelsidiga konfidensintervall (KI) för medelvärden och för andelar. Funktionen ska heta `my_conf(x=,alpha=)`. Argumentet `x=` ska vara en numerisk vektor eller en vektor med bara ettor och nollor. `alpha=` anger konfidensgrad, där `alpha=0.05` ger 95 % KI. Inga inbyggda funktioner för att beräkna KI i R får användas utan de ska "räknas för hand". KI för medelvärden ska beräknas enligt:

$$\bar{x} \pm z_* \cdot \frac{s}{\sqrt{n}}$$

Där  $\bar{x}$  är stickprovets medelvärde,  $s$  är stickprovets standardavvikelse och  $n$  är antalet observationer.  $z_*$  är lämplig kvantil från standardiserade normalfördelningen ( $N(x, \mu = 0, \sigma = 1)$ ). T.ex. så är  $z_* = 1.959964$  korrekt för dubbelsidigt KI med `alpha=0.05`. KI för andelar ska beräknas enligt:

$$p \pm z_* \cdot \sqrt{\frac{p(1-p)}{n}}$$

Där  $p$  stickprovsandelen,  $z_*$  och  $n$  är samma som ovan. Om `x` bara består av ettor och nollor så ska KI för andelar beräknas, annars ska KI för medelvärde beräknas. Om `x` inte är numerisk eller heltal så ska funktionen avbrytas med `stop()` och skriva ut följande meddelande "x is not numeric". KI ska returneras som en numerisk vektor. Se testfallen för exempel på hur funktionen ska fungera.

```
a<-rnorm(100,3,4)
b<-rbinom(n = 300,size = 1,prob = 0.4)
my_conf(x = a,alpha = 0.05)

[1] 1.92884 3.69114

my_conf(x = a,alpha = 0.001)

[1] 1.33065 4.28933

my_conf(x = b,alpha = 0.1)

[1] 0.353477 0.446523

my_conf(x = b,alpha = 0.01)

[1] 0.327145 0.472855
```

#### 5. Statistik och funktioner (1p)

- (a) Nu ska funktionen `list_stats(x=)` skapas. Argumentet `x` ska vara en lista där varje element är numerisk matris eller numerisk `data.frame`. Funktionen ska gå igenom alla element i `x` och använda funktionen `summary()` på varje kolumn i matrisen/`data.frame`. Resultatet ska sedan sparas i en lista av samma längd som `x`. Se testfallet nedan för hur funktionen ska fungera. **1p**

```
b<-list(trees,matrix(rnorm(400),100,4))
list_stats(x = b)
```

```
[[1]]
```

	Girth	Height	Volume
Min.	: 8.3	Min. :63	Min. :10.2
1st Qu.:	11.1	1st Qu.:72	1st Qu.:19.4
Median :	12.9	Median :76	Median :24.2
Mean :	13.2	Mean :76	Mean :30.2
3rd Qu.:	15.2	3rd Qu.:80	3rd Qu.:37.3
Max. :	20.6	Max. :87	Max. :77.0

```
[[2]]
```

	V1	V2	V3	V4
Min.	:-2.7582	Min. : -1.8492	Min. : -2.4435	Min. : -2.9243
1st Qu.:	-0.5920	1st Qu.: -0.5165	1st Qu.: -0.4890	1st Qu.: -0.7165
Median :	0.0219	Median : 0.0723	Median : 0.0256	Median : 0.1887
Mean :	-0.0428	Mean : 0.0951	Mean : 0.0416	Mean : 0.0738
3rd Qu.:	0.5576	3rd Qu.: 0.6581	3rd Qu.: 0.6536	3rd Qu.: 0.8299
Max. :	2.2305	Max. : 2.1659	Max. : 2.5458	Max. : 1.9985

## 6. Statistik och grafik (3p)

- (a) Läs in det interna datamaterialet `PlantGrowth`. Gör sedan ett two sample t-test för variabeln `weight`, där du jämför skillanden i medelvärde för grupperna `trt1` och `trt2`. Skriv även en kod som väljer ut konfidensintervallet för medelvärdesskillanden från t-test-objektet och spara det i variabeln `my_conf`. **1p**
- (b) Utgå från det interna datamaterialet `mpg` som kommer med paketet `ggplot2`. Använd funktioner i `ggplot2` för att: Skapa en scatter plot där `cty` är på x-axel och `displ` är på y-axeln. Scatter plotten ska vara uppdelad i *radvisa* paneler efter variabeln `drv` (detta ger 3 subplottar). Varje panelplot ska ha en linjär regressionslinje genom punktsvärmen. **1.5p**
- (c) Skapa valfri barplot över variabeln `drv` i datasetet `mpg`. Tex skulle grafen kunna se ut enligt nedan: **0.5p**

