Labb 0

Sebastijan Babic & Esbjörn Runesson

XXXX–XX–XX

Uppgift 1

Definiera en variabel x som har värdet $\pi + 1$. Använd värdet på denna variabel, avrundat till 3 decimaler, mitt i en mening. Skriv t.ex. "Värdet av $\pi + 1$ är ca ...". Hint: skriv ?round i Console för information om denna funktion, och notera att R har en inbyggd konstant pi.

```
# skapa variabel x med pi + 1
x <- pi + 1
x_rounded <- round(x, 3)

paste("Värdet av pi + 1 är ca:", x_rounded)
```

```
## [1] "Värdet av pi + 1 är ca: 4.142"
```

Uppgift 2

Definiera en funktion med passande namn, som tar som argument en radie och en höjd, och returnerar arean för en cylinder med samma radie och höjd. Arean ska inkludera botten och toppen på cylindern. Googla formeln som ska användas om du inte kommer ihåg den eller orkar härleda den! Använd funktionen för att räkna ut arean för en cylinder som har höjd 7 meter och radie 3 meter, samt en som har har höjden 8 meter och radie 29 meter.

```
cylinder_area <- function (r, h) {
  area <- 2 * pi * r * (r + h)
  area_rounded <- round(area, 1)
  return(area_rounded)
}

paste("Arean av en cylinder med radie 3 och höjd 7 är:", cylinder_area(3, 7))
```

```
## [1] "Arean av en cylinder med radie 3 och höjd 7 är: 188.5"
```

```
paste("Arean av en cylinder med radie 29 och höjd 8 är:", cylinder_area(29, 8))
```

```
## [1] "Arean av en cylinder med radie 29 och höjd 8 är: 6741.9"
```

Uppgift 3

Använd funktionen du just definierade för att skapa en plot av en cylinders area då radien varierar mellan 0.1 och 4 längdenheter. Höjden kan vara fix på 1 längdenhet. Plotten bör ha passande rubriker på axlarna och grafen till funktionen bör vara en linje, inte punkter. Förse plotten med ett nummer och en beskrivande text, t.ex. “Diagram 1: En cylinders area som funktion av radie.

```r
radius_values <- seq(0.1, 4, by = 0.1)
areas <- sapply(radius_values, cylinder_area, h = 1) # applicera funktion på list av vektorerna, radius

plot(radius_values, areas, type = "l", col = "blue", lwd = 2,
    xlab = "Radie (längdenheter)", ylab = "Area (kvadratenheter)",
    main = "En cylinders area som funktion av radie")
```

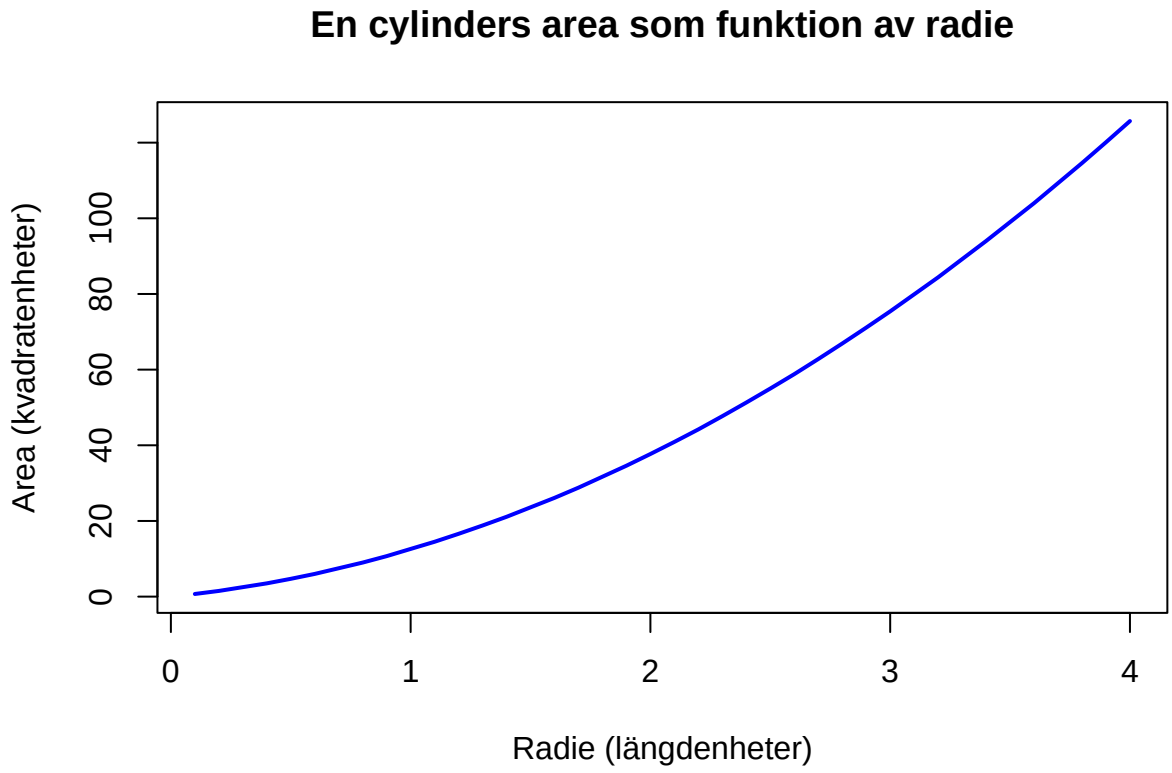

Figure 1: Diagram som visar en cylinders area som funktion av radie

Uppgift 4

Deluppgift 4a

Definiera en funktion medelv1 som räknar ut medelvärdet av elementen i en vektor x genom att använda funktionerna sum och length. Testa denna funktion på någon vektor du hittar på, och jämför resultatet med det du får när du använder den inbyggda funktionen mean på samma vektor.

```r
medelv1 <- function(x) {
  medelv1 <- sum(x) / length(x) # calculate mean
  return(medelv1)
}

x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
paste("Medelvärdet av elementen i en vektor x är ", medelv1(x))
```

```
## [1] "Medelvärdet av elementen i en vektor x är  5.5"
```

```r
paste("Medelvärdet av elementen i en vektor x med den inbyggda funktionen är ", mean(x))
```

```
## [1] "Medelvärdet av elementen i en vektor x med den inbyggda funktionen är  5.5"
```

Deluppgift 4b

Definiera en funktion medelv2 som räknar ut medelvärdet av elementen i en vektor x genom att använda en for-loop. Testa denna funktion på någon vektor du hittar på, och jämför resultatet med det du får när du använder den inbyggda funktionen mean på samma vektor.

```r
medelv2 <- function(x) {
    sum <- 0 # init sum
    for (i in 1:length(x)) { # loop through x and sum up all elements in x
        sum <- sum + x[i]
    }
    medelv2 <- sum / length(x)  # calculate mean
    return(medelv2)
}

medelv2(x)
```

```
## [1] 5.5
```

Uppgift 5

Här är years en vektor av år efter en viss tidpunk, och gdp_capita är BNP (bruttonationalprodukt - gross domestic product på engelska) per capita i ett påhittat land. Vad funktionen rnorm(20) gör här är att till punkterna på den räta linjen med intercept 10 och lutning 0.3 lägga till ett normalfördelat brus med väntevärde 0 och standardavvikelse 1 . Vad detta faktiskt betyder kommer ni få veta mer om senare i kursen. Funktionen set.seed gör här att det normalfördelade bruset, dvs de normalfördelade slumptal som läggs till på den räta linjen, är desamma varje gång koden körs.

```r
set.seed(12345)

years <- 0:19
gdp_capita <- 10 + 0.3 * years + rnorm(20)
```

Uppgift 5a och 5b

Kör koden ovan och skapa en plot med punkter, med years på x-axeln och gnp_capita på y-axeln. Se till att ha ordentliga rubriker på axlarna.

Skriv ?abline i Console i RStudio för att se hur du kan använda denna funktion för att i samma plot som ovan rita in en rät linje med intercept 10 och lutning 0.3 . Välj gärna en annan färg än svart på denna linje - hur man gör det kan du se bland exemplena i hjälpen för funktionen abline. Prova ändra på värdet i set.seed till något annat heltal än 12345 . Vad händer om du nu skapar plotten på nytt?

```
par(mfrow = c(1, 2))

plot(x = years, y = gdp_capita, xlab = "År", ylab = "BNP per capita", main = "BNP per capita över tid")
abline(a = 10, b = 0.3, col = "red")

set.seed(11111)
gdp_capita <- 10 + 0.3 * years + rnorm(20) # regenerate gdp_capita with new seed

plot(x = years, y = gdp_capita, xlab = "År", ylab = "BNP per capita", main = "BNP per capita över tid")
abline(a = 10, b = 0.3, col = "red")
```

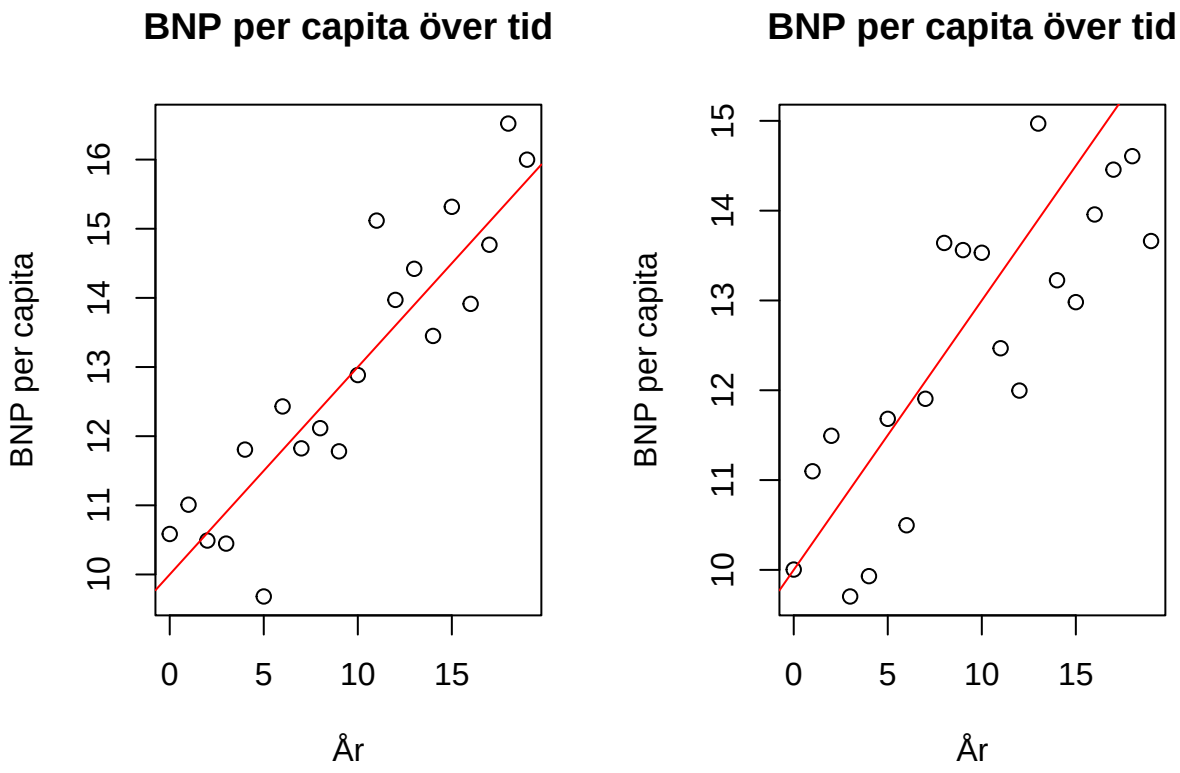

Figure 2: Plot med år på x-axeln och BNP per capita på y-axeln

```
par(mfrow = c(1, 1))
```