Data analysis with R final exam 2022

2022-07-04

1.

```
tmp <- c(4, 6, 3)
```

Create the vectors

(a) $(4, 6, 3, 4, 6, 3, \ldots, 4, 6, 3)$ where there are 10 occurrences of 4.

```
rep(c(4,6,3), times=10)
```

```
##  [1] 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3
```

(b) $(4, 4, \ldots, 4, 6, 6, \ldots, 6, 3, 3, \ldots, 3)$ where there are 10 occurrences of 4, 20 occurrences of 6 and 30 occurrences of 3.

```
cat(rep(4, each=10), rep(6, each=20), rep(3, each=30))
```

```
## 4 4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

2.

Execute the following lines which create two vectors of random integers which are chosen with replacement from the integers $0, 1, \ldots, 999$. Both vectors have length 250.

```
xVec <- sample(0:999, 250, replace=T)
yVec <- sample(0:999, 250, replace=T)
```

(a) Create the vector $(y_2 - x_1, \cdots, y_n - x_{n-1})$.

```
tmp <- yVec[-1]-xVec[-250]
```

(b) Pick out the values in yVec which are > 600.

```
tmp <- yVec[yVec > 600]
```

(c) What are the index positions in yVec of the values which are > 600?

```
which(yVec > 600)
```

```
##  [1]   9  12  14  15  16  21  24  25  29  31  32  36  40  42  48  51  52  57  59
## [20]  64  65  68  70  73  75  76  80  88  91  92  93  96  97 102 107 108 110 111
## [39] 112 113 114 117 118 119 124 132 135 139 152 153 157 166 167 169 171 172 173
## [58] 174 175 176 180 186 189 190 193 194 197 199 200 202 204 205 209 210 213 214
## [77] 215 216 218 219 220 221 222 223 225 226 231 232 233 235 237 241 242 244 249
## [96] 250
```

(d) Sort the numbers in the vector xVec in the order of increasing values in yVec.

(e) Pick out the elements in yVec at index positions $1, 4, 7, 10, 13, \cdots$

```
idx <- seq(1,250, by=3)
tmp <- yVec[idx]
```

3.

By using the function cumprod and other functions to calculate:

$$1 + \frac{2}{3} + \left(\frac{2}{3}\frac{4}{5}\right) + \left(\frac{2}{3}\frac{4}{5}\frac{6}{7}\right) + \cdots + \left(\frac{2}{3}\frac{4}{5}\cdots\frac{38}{39}\right)$$

```
x <- seq(2,38, by=2) / seq(3,39, by=2)
result <- 1 + sum(cumprod(x))
result
```

```
## [1] 6.976346
```

4.

For this problem we'll use the (built-in) dataset state.x77.

```
data(state)
state.x77 <- as.data.frame(state.x77)
head(state.x77)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
## California      21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

a. Find out how many states have an income of less than 4300.

```
length(state.x77$Income < 4300)
```

```
## [1] 50
```

b. Find out which is the state with the highest income.

```
rownames(state.x77[which(state.x77$Income==max(state.x77$Income)),])
```

```
## [1] "Alaska"
```

c. Add a variable to the data frame which should categorize the level of illiteracy: $[0, 1)$ is low, $[1, 2)$ is some, $[2, \infty)$ is high.

```
library(dplyr)
state.x77 <- state.x77 %>% mutate(Level =
            ifelse(Illiteracy<1, "low",
                ifelse(Illiteracy<2, "some", "high")))
head(state.x77)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
## California      21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
##            Level
## Alabama     high
## Alaska      some
## Arizona     some
## Arkansas    some
## California  some
## Colorado     low
```

d. Find out which state with low illiteracy, has the highest income, and what that income is.

```
state.tmp <- state.x77 %>% filter(Level=="low")
max(state.tmp$Income)
```

```
## [1] 5299
```

```
rownames(state.tmp[which(state.tmp$Income==max(state.tmp$Income)),])
```

```
## [1] "Maryland"
```

5.

Simulate 1,000 observations from (X_1, X_2) which follow the uniform distribution over the square $[0, 1] \times [0, 1]$.

```
x1 <- runif(1000, 0,1)
x2 <- runif(1000, 0,1)
```

a. Get an approximation of the probability that the distance between (X_1, X_2) and the nearest edge is less than 0.25.

```
length(x1[x1<0.25 | 1-x1<0.25 | x2<0.25 | 1-x2<0.25])/1000
```

```
## [1] 0.74
```

b. The same question for the distance to the nearest vertex.

```
length(x1[x1^2+x2^2<0.25^2 | (1-x1)^2+x2^2<0.25^2 |
    x1^2+(1-x2)^2<0.25^2 | (1-x1)^2+(1-x2)^2<0.25^2])/1000
```

```
## [1] 0.203
```

6.

A discrete random variable X has probability mass function

x	0	1	2	3	4
$p(x)$	0.1	0.2	0.2	0.2	0.3

Generate a random sample of size 1000 from the distribution of X using the R `sample()` function. Construct a relative frequency table and compare the empirical with the theoretical probabilities.

```
x <- c(0, 1,1, 2,2, 3,3, 4,4,4)
samp <- sample(x, size=1000, replace=T)
res <- data.frame(
    x_equal_0 = c(0.1, length(samp[samp==0])/1000),
    x_equal_1 = c(0.2, length(samp[samp==1])/1000),
    x_equal_2 = c(0.2, length(samp[samp==2])/1000),
    x_equal_3 = c(0.2, length(samp[samp==3])/1000),
    x_equal_4 = c(0.3, length(samp[samp==4])/1000)
)
rownames(res) <- c("theoretical_Pr", "empirical_Pr")
res
```

```
##                x_equal_0 x_equal_1 x_equal_2 x_equal_3 x_equal_4
## theoretical_Pr     0.100     0.200     0.200     0.200     0.300
## empirical_Pr       0.105     0.178     0.198     0.216     0.303
```

7.

Mortality rates per 100,000 from male suicides for a number of age groups and a number of countries are given in the following data frame.

```
suicrates <- tibble(Country = c('Canada', 'Israel', 'Japan', 'Austria', 'France', 'Germany',
'Hungary', 'Italy', 'Netherlands', 'Poland', 'Spain', 'Sweden', 'Switzerland', 'UK', 'USA'),
Age25.34 = c(22,  9, 22, 29, 16, 28, 48,  7,  8, 26,  4, 28, 22, 10, 20),
Age35.44 = c(27, 19, 19, 40, 25, 35, 65,  8, 11, 29,  7, 41, 34, 13, 22),
Age45.54 = c(31, 10, 21, 52, 36, 41, 84, 11, 18, 36, 10, 46, 41, 15, 28),
Age55.64 = c(34, 14, 31, 53, 47, 49, 81, 18, 20, 32, 16, 51, 50, 17, 33),
Age65.74 = c(24, 27, 49, 69, 56, 52, 107, 27, 28, 28, 22, 35, 51, 22, 37))
```

a. Transform `suicrates` into *long* form.

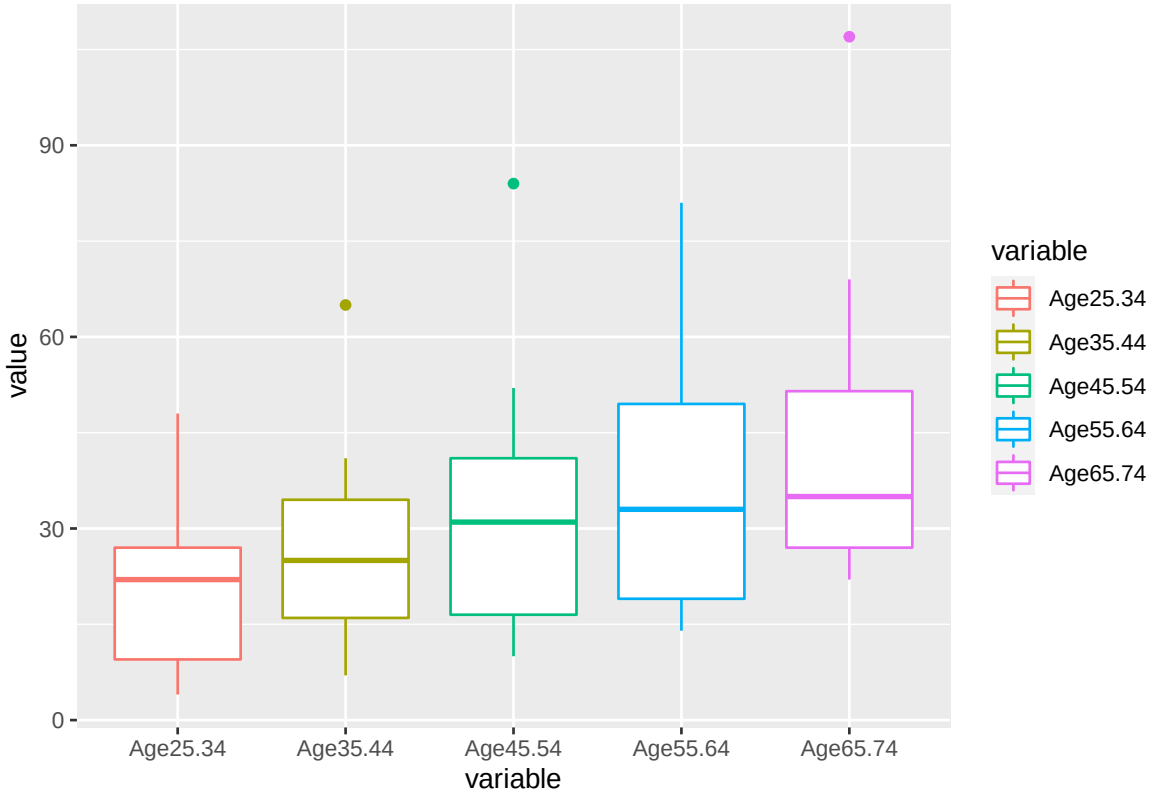
```r
library(reshape)
longdata <- melt(as.data.frame(suicrates), id="Country")
longdata
```

```
##        Country variable value
## 1       Canada Age25.34    22
## 2       Israel Age25.34     9
## 3        Japan Age25.34    22
## 4      Austria Age25.34    29
## 5       France Age25.34    16
## 6      Germany Age25.34    28
## 7      Hungary Age25.34    48
## 8        Italy Age25.34     7
## 9  Netherlands Age25.34     8
## 10      Poland Age25.34    26
## 11       Spain Age25.34     4
## 12      Sweden Age25.34    28
## 13 Switzerland Age25.34    22
## 14          UK Age25.34    10
## 15         USA Age25.34    20
## 16      Canada Age35.44    27
## 17      Israel Age35.44    19
## 18       Japan Age35.44    19
## 19     Austria Age35.44    40
## 20      France Age35.44    25
## 21     Germany Age35.44    35
## 22     Hungary Age35.44    65
## 23       Italy Age35.44     8
## 24 Netherlands Age35.44    11
## 25      Poland Age35.44    29
## 26       Spain Age35.44     7
## 27      Sweden Age35.44    41
## 28 Switzerland Age35.44    34
## 29          UK Age35.44    13
## 30         USA Age35.44    22
## 31      Canada Age45.54    31
## 32      Israel Age45.54    10
## 33       Japan Age45.54    21
## 34     Austria Age45.54    52
## 35      France Age45.54    36
## 36     Germany Age45.54    41
## 37     Hungary Age45.54    84
## 38       Italy Age45.54    11
## 39 Netherlands Age45.54    18
## 40      Poland Age45.54    36
## 41       Spain Age45.54    10
## 42      Sweden Age45.54    46
## 43 Switzerland Age45.54    41
## 44          UK Age45.54    15
## 45         USA Age45.54    28
## 46      Canada Age55.64    34
## 47      Israel Age55.64    14
## 48       Japan Age55.64    31
```

```
## 49     Austria Age55.64    53
## 50      France Age55.64    47
## 51     Germany Age55.64    49
## 52     Hungary Age55.64    81
## 53       Italy Age55.64    18
## 54 Netherlands Age55.64    20
## 55      Poland Age55.64    32
## 56       Spain Age55.64    16
## 57      Sweden Age55.64    51
## 58 Switzerland Age55.64    50
## 59          UK Age55.64    17
## 60         USA Age55.64    33
## 61      Canada Age65.74    24
## 62      Israel Age65.74    27
## 63       Japan Age65.74    49
## 64     Austria Age65.74    69
## 65      France Age65.74    56
## 66     Germany Age65.74    52
## 67     Hungary Age65.74   107
## 68       Italy Age65.74    27
## 69 Netherlands Age65.74    28
## 70      Poland Age65.74    28
## 71       Spain Age65.74    22
## 72      Sweden Age65.74    35
## 73 Switzerland Age65.74    51
## 74          UK Age65.74    22
## 75         USA Age65.74    37
```

b. Construct side-by-side box plots for the data from different age groups, and comment on what the graphic tells us about the data.

```
library(ggplot2)
ggplot(longdata, aes(x=variable, y=value, color=variable)) +
    geom_boxplot()
```



As seen in the image, the lowest suicide rate is in the 25-34 age group and the highest in the 65-74 age group. As the age increases, the suicide rate is gradually increasing.

8.

The steam data in the `MASS` package has a nonlinear regression model,

$$P = \alpha \exp\left\{\frac{\beta t}{\gamma + t}\right\} + \varepsilon$$

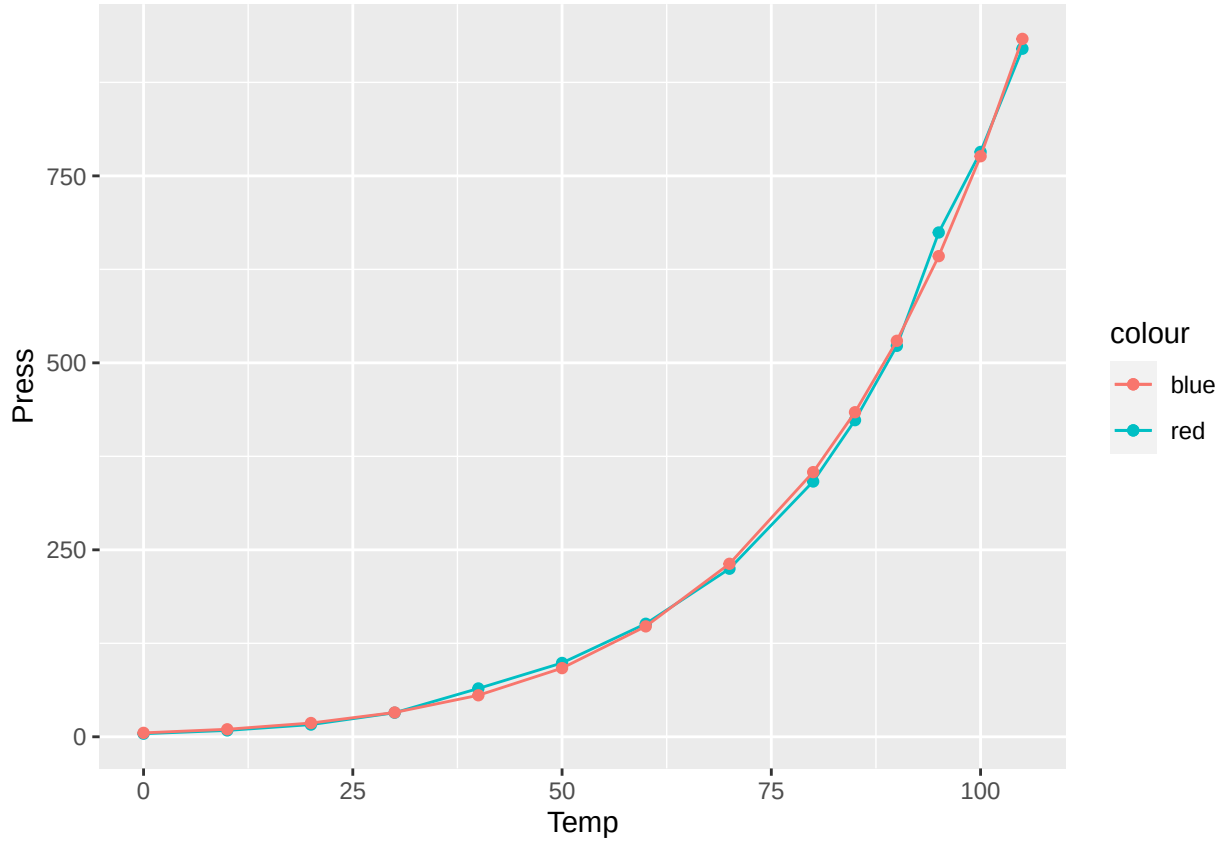
Fit the model with `nls()` function and find the fitted values, using initial value $\alpha = 5, \beta = 20, \gamma = 200$. Plot them with the data points on the original scale.

```
library(MASS)
data(steam)
f <- function(t, a,b,g){
    return (a*exp(b*t/(g+t)))
}
fitted.model <- nls(Press~f(Temp, a,b,g),
        data=steam, start=list(a=5,b=20,g=200))
summary(fitted.model)
```

```
##
## Formula: Press ~ f(Temp, a, b, g)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
```

```
## a    5.267      2.275   2.316  0.04088 *
## b   19.722      4.706   4.191  0.00151 **
## g  294.995    127.219   2.319  0.04066 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.5 on 11 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 6.172e-06
```

```
steam <- steam %>% mutate(FittedModel =
                predict(fitted.model, steam))
ggplot(steam) + geom_point(aes(Temp, Press, color="red"))+
        geom_line(aes(Temp, Press, color="red")) +
        geom_point(aes(Temp, FittedModel, color="blue"))+
        geom_line(aes(Temp, FittedModel, color="blue"))
```



"red" is Press, while "blue" is FittedModel.