Solution Intro-3

Dr. Osama Mahmoud

05/07/2019

Exercise 1:

a) consider the following data - similar to the example used for `practical Intro-1` - to generate a data frame named `DT`:

id	ht	wt	gender
1	155	80	m
2	152	85	m
3	164	45	f
4	175	69	m
5	193	86	f
6	203	110	f
7	190	106	f
8	183	96	m
9	155	90	f
10	169	89	m

```
DT = data.frame(id = 1:10,
                ht=c(155, 152, 164, 175, 193, 203, 190, 183, 155, 169),
                wt=c(80, 85, 45, 69, 86, 110, 106, 96, 90, 89),
                gender=c("m", "m", "f", "m", "f", "f", "f", "m", "f", "m"))
DT
```

```
##    id  ht  wt gender
## 1   1 155  80      m
## 2   2 152  85      m
## 3   3 164  45      f
## 4   4 175  69      m
## 5   5 193  86      f
## 6   6 203 110      f
## 7   7 190 106      f
## 8   8 183  96      m
## 9   9 155  90      f
## 10 10 169  89      m
```

b) Add a new categorical variable `bmi.grp` to the data frame which is defined as follows:

- bmi $\leq 18.5 \rightarrow$ underweight
- $18.5 <$ bmi $\leq 25 \rightarrow$ normal
- $25 <$ bmi $\leq 30 \rightarrow$ overweight
- $30 <$ bmi $\rightarrow$ obesity

```
DT$bmi = DT$wt/(DT$ht/100)^2
DT$bmi.grp = ifelse(DT$bmi <= 18.5, "underweight", "obesity")
DT$bmi.grp = ifelse(DT$bmi > 18.5 & DT$bmi <= 25, "normal", DT$bmi.grp)
DT$bmi.grp = ifelse(DT$bmi > 25 & DT$bmi <= 30, "overweight", DT$bmi.grp)
```

c) Generate a vector `z` consisting of the first 30 elements of the Fibonacci series. By definition, the first two elements of this series are 0 and 1. All further elements are calculated as sum of the preceding two elements, so $z = 0, 1, 1, 2, 3, 5, \dots$

```
# Manually set the first two elements
z = c(0,1)
for(i in 2:29){
  z[i+1] = z[i] + z[i-1]
}
z
```

```
##  [1]      0      1      1      2      3      5      8     13     21     34
## [11]     55     89    144    233    377    610    987   1597   2584   4181
## [21]   6765  10946  17711  28657  46368  75025 121393 196418 317811 514229
```

Exercise 2:

a) load the internal R data set `mtcars` and view its help page to find out abouts its varaible description.

```
data(mtcars)
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
?mtcars
```

b) Fit a regression model of "Number of car cylinders"" on "Miles per gallon" using the function `lm(mpg ~ cyl, data=mtcars)` and assign the `Model` for that object. Then show `summary` of that object.

```
Model = lm(mpg ~ cyl, data=mtcars)
summary(Model)
```

```
## 
## Call:
## lm(formula = mpg ~ cyl, data = mtcars)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.9814 -2.1185  0.2217  1.0717  7.5186 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  37.8846     2.0738   18.27  < 2e-16 ***
## cyl          -2.8758     0.3224   -8.92 6.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```
## Residual standard error: 3.206 on 30 degrees of freedom
## Multiple R-squared:  0.7262, Adjusted R-squared:  0.7171
## F-statistic: 79.56 on 1 and 30 DF,  p-value: 6.113e-10
```

c) Write a function to extract the effect estimate, standard error and the p-value from a linear regression model. Call the function `beta_se`. Hint: use the regression model object as the input and extract summary(model)$coefficient.

```r
# to explore what names can be extracted from the summary object.
names(summary(Model))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
# The function
beta_se = function(Simple.Reg.model){
  Summary.model = summary(Simple.Reg.model)
  Out = Summary.model$coefficients[2,c("Estimate", "Std. Error", "Pr(>|t|)")]
  return(Out)
}
# Implement the `beta_se` function
beta_se(Model)
```

```
##      Estimate    Std. Error      Pr(>|t|)
## -2.875790e+00  3.224089e-01  6.112687e-10
```

d) Use the `beta_se` function to extract the effect estimates, standard errors and p-values of the regression models of (1)“Displacement” on “Miles per gallon” and (2) “Rear axle ratio” on “Miles per gallon”.

```r
Model1 = lm(mpg ~ disp, data = mtcars)
beta_se(Model1)
```

```
##      Estimate    Std. Error      Pr(>|t|)
## -4.121512e-02  4.711833e-03  9.380327e-10
```

```r
Model2 = lm(mpg ~ drat, data = mtcars)
beta_se(Model2)
```

```
##     Estimate   Std. Error     Pr(>|t|)
## 7.6782326020 1.5067051076 0.0000177624
```