Q5.

- Print the first 3 observations and last 4 variables.

```
heartAttackdf <- as.data.frame(read.csv("./data/heart-attack.csv"))

haCols <- ncol(heartAttackdf)
heartAttackdf[1:3, (haCols-4):haCols]
```

```
##   Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1          Urban            110.89 17.6                      0
## 2          Rural             69.04 35.9 formerly smoked      0
## 3          Urban            210.95 50.1                      0
```

- Print the first 3 observations and `age` and `work_type`.

```
heartAttackdf[1:3,][c('age', 'work_type')]
```

```
##   age work_type
## 1   8   Private
## 2  70   Private
## 3  47   Private
```

- Print the first 3 observations and 1st, 4th, and 7th variables.

```
heartAttackdf[1:3, c(1,4,7)]
```

```
##      id hypertension work_type
## 1 16523            0   Private
## 2 56543            0   Private
## 3 32257            0   Private
```

- How many married people had a stroke?

```
# had a stroke
hadStroke <- sum(heartAttackdf$stroke) # 783

# didn't have a stroke
noStroke <- sum(heartAttackdf$stroke==0) # 4,2617

# checking total equals the total number of rows
nrow(heartAttackdf) == hadStroke + noStroke # returns TRUE
```

```
## [1] TRUE
```

```
# Married people with a stroke
nrow(heartAttackdf[heartAttackdf$stroke==1 & heartAttackdf$ever_married=="Yes",]) # 703
```

```
## [1] 703
```

- How many people below the age of 20 had a stroke?

```
nrow(heartAttackdf[heartAttackdf$stroke==1 & heartAttackdf$age < 20,]) # 2
```

- How many `private` and `self-employed` people had a stroke?

```
nrow(heartAttackdf[heartAttackdf$work_type == "private" &
                     heartAttackdf$work_type == "self-employed" &
                     heartAttackdf$stroke == 1]) # 43400
```

- Presuming the data frame in which your data is stored is called `myDF`, explain why the output of `myDF[c(1, 2)]` and `myDF[ ,c(1, 2)]` is the same.

```
heartAttackdf[c(1,2)]
heartAttackdf[ , c(1, 2)] # selects all rows of column 1 and 2
```

The command `df[c(1,2)]` selects column 1 and column 2, and is equivalent to `df[1]`, `df[2]` or `df['id']`, `df['gender']`.

The second command, `df[, c(1,2)]` produces the same output because when a comma is included, the first parameter refers to the rows and the second to the columns. When the first parameter is left blank (which is what we have here), then all rows selected by default.

Q6.

Import the file `fish.csv` (information on specific types of fish) and answer the following questions using `dplyr`:

- Select all the columns that start with `cestode`

```
library("dplyr")
library("magrittr")
fish <- as.data.frame(read.csv("./data/fish.csv"))

fish %>%
  select(starts_with("cestcode")) # 1,800 rows
```

- Select all observations where `wet_weight` is greater than 0.2

```
fish %>%
  filter(wet_weight > 0.2)
```

- Select all observations where the `coastal_ecological_area` is Lake Michigan

```
fish %>%
  filter(coastal_ecological_area == "Lake Michigan")
```

- Select all observations where `sex` is Male **AND** `state_name` is Mississippi

```
fish %>%
  filter(sex=="Male" & state_name=="Mississippi")
```

- Select all observations where `sex` is Male **OR** `state_name` is Mississippi

```
fish %>%
  filter(sex=="Male" || state_name=="Mississippi")
```

- Create a new variable called `large_fish` that is TRUE if a fish is over 10 oz.

```
fish %>%
  mutate(large_fish = wet_weight > 10)
```

- Create a new variable `parasites` that is TRUE if a fish has more than 1 `unidentified_organism` in it and a `wet_weight` of less than 0.5 oz

```
fish %>%
  mutate(parasites = (unidentified_organism > 1 & wet_weight < 0.5))
```

- Summarize (mean, median, max, min) the length of the fish by `state_name`

```
fish %>%
  group_by(state_name) %>%
  summarize(meanLength = mean(length),
            medianLength = median(length), maxLength = max(length), minLength = min(length))
```

Interesting! Alabama and Texas have really long fish on average, but Texas has the largest possible fish by a mile.