Exercise 1

Let us first create the three vectors.

```
x <- sample(100)
y <- sample(100)
z <- sample(letters, 100, replace = TRUE)
```

Now we can start to extract elements from `x` according to the exercises.

(a) We can extract every fifth value in two ways. Either by logical subsetting or by position.

```
x[c(TRUE, FALSE, FALSE, FALSE, FALSE)]

#  [1] 32 85 10 90 72  1 88 64 76 30 71 77 91 51 61
# [16] 75 21 95 12 65

x[seq(1, 100, 5)]

#  [1] 32 85 10 90 72  1 88 64 76 30 71 77 91 51 61
# [16] 75 21 95 12 65
```

(b) We have seen in the lecture that we can omit elements by using negative position values.

```
x[-30:-10] # Create a sequence of values from -30 to -10.

#  [1] 32  6 45 50  5 85 93 52 80 88 69 78 92 68 64
# [16] 56 83 20 55 76 46 15 26 34 30 44 33 58 59 71
# [31] 43 36  7 67 77 25 94 53 96 91  2 79 89 11 51
# [46] 23  3 84 28 61 14 49 60 38 75 66 47 13 31 21
# [61] 87 24 35 74 95 57 81 48 18 12 41 39 37 97 65
# [76] 54 42 63 22
```

(c) The sequence is the cumulative sum of all values from 1 to 13. We can use this sequence to extract the elements from x.

```
cumsum(1:13)

#  [1]  1  3  6 10 15 21 28 36 45 55 66 78 91

x[cumsum(1:13)]

#  [1] 32 45 85 17  9 72 19 64 34 67 51 47 12
```

(d) We have to use logical subsetting for this. The idea is to make a comparison, if `y` is smaller than `x`. Since such comparisons are done elementwise for two vectors we can use the resulting logical vector for answering the question.

```
x[y < x]

#  [1]  32  45  50  52  80  82  90  73  70  72  98
# [12]  99  86 100  62  88  78  92  64  83  55  76
# [23]  58  71  43  67  77  94  96  91  79  89  11
# [34]  23  84  75  66  47  31  87  24  35  74  95
# [45]  48  41  97  65
```

(e) This is very much like the previous exercise. We have to create a logical vector. We can use `%in%` to check for every element in `z` if it is either `"c"`, `"r"`, or `"x"`.

```
x[z %in% c("c", "r", "x")]

# [1]  6 17 99 26 30 44  7 94  3
```

(f) Fort his exercise we have to use the `&` (intersection) operator. We create two logical vectors (those from (d) and (e)) and connect them.

```
x[(y < x) & (z %in% c("c", "r", "x"))]

# [1] 99 94
```

(g) The same as in (f) but this time we have to use the `|` (union) operator.

```
x[(y < x) | (z %in% c("c", "r", "x"))]

#  [1]  32   6  45  50  52  80  17  82  90  73  70
# [12]  72  98  99  86 100  62  88  78  92  64  83
# [23]  55  76  26  30  44  58  71  43   7  67  77
# [34]  94  96  91  79  89  11  23   3  84  75  66
# [45]  47  31  87  24  35  74  95  48  41  97  65
```

(h) The position of the largest element in `x` can be detected using `which.max`.

```
which.max(x)

# [1] 29
```

If you have trouble to comprehend the solutions I recommend that you dissect the solutions piece by piece. Take the expressions in the brackets and execute them alone. If they are assembled like in exercise (f) and (g) disassemble the expressions further. Also go through the lecture again and execute the examples. If you are not sure, what exactly happens, disassemble the examples. Try around yourself.

Exercise 2

We can not extract single elements cause there is no meaningful data structure this data could take. For example, the elements in the iris data set are of different types, so we can not put them in a single vector.

```
iris[5,4]

# [1] 0.2

iris[4,5]

# [1] setosa
# Levels: setosa versicolor virginica
```

A data frame is also not possible. A data frame is rectangular. What happens to the diagonal elements in the fifth row + fifth column and fourth row + fourth column? A list would theoretically be possible but the ordering of the elements would not be defined. This is too ambiguous for being a reliable tool in programming.
In the following we have two possibilities to extract the desired information: Either using brackets or using `subset`. I will show you both ways for the first two problems. I think that after exercise 1 the solutions are more or less self-explanatory so I will omit further explanations. For using brackets you have to remember to put the comma!

(a)
```
subset(USArrests, Murder >= 10)

#                Murder Assault UrbanPop Rape
# Alabama          13.2     236       58 21.2
# Alaska           10.0     263       48 44.5
# Florida          15.4     335       80 31.9
# Georgia          17.4     211       60 25.8
# Illinois         10.4     249       83 24.0
# Louisiana        15.4     249       66 22.2
# Maryland         11.3     300       67 27.8
# Michigan         12.1     255       74 35.1
# Mississippi      16.1     259       44 17.1
# Nevada           12.2     252       81 46.0
# New Mexico       11.4     285       70 32.1
# New York         11.1     254       86 26.1
# North Carolina   13.0     337       45 16.1
# South Carolina   14.4     279       48 22.5
# Tennessee        13.2     188       59 26.9
# Texas            12.7     201       80 25.5
```

```
USArrests[USArrests$Murder >= 10,]

#                Murder Assault UrbanPop Rape
# Alabama          13.2     236       58 21.2
# Alaska           10.0     263       48 44.5
# Florida          15.4     335       80 31.9
# Georgia          17.4     211       60 25.8
# Illinois         10.4     249       83 24.0
# Louisiana        15.4     249       66 22.2
# Maryland         11.3     300       67 27.8
# Michigan         12.1     255       74 35.1
# Mississippi      16.1     259       44 17.1
# Nevada           12.2     252       81 46.0
# New Mexico       11.4     285       70 32.1
# New York         11.1     254       86 26.1
# North Carolina   13.0     337       45 16.1
# South Carolina   14.4     279       48 22.5
# Tennessee        13.2     188       59 26.9
# Texas            12.7     201       80 25.5
```

(b)

```
subset(USArrests, Rape >= 20 & Rape <= 40)

#                Murder Assault UrbanPop Rape
# Alabama          13.2     236       58 21.2
# Arizona           8.1     294       80 31.0
# Colorado          7.9     204       78 38.7
# Florida          15.4     335       80 31.9
# Georgia          17.4     211       60 25.8
# Hawaii            5.3      46       83 20.2
# Illinois         10.4     249       83 24.0
# Indiana           7.2     113       65 21.0
# Louisiana        15.4     249       66 22.2
# Maryland         11.3     300       67 27.8
# Michigan         12.1     255       74 35.1
# Missouri          9.0     178       70 28.2
# New Mexico       11.4     285       70 32.1
# New York         11.1     254       86 26.1
# Ohio              7.3     120       75 21.4
# Oklahoma          6.6     151       68 20.0
# Oregon            4.9     159       67 29.3
# South Carolina   14.4     279       48 22.5
# Tennessee        13.2     188       59 26.9
# Texas            12.7     201       80 25.5
# Utah              3.2     120       80 22.9
# Virginia          8.5     156       63 20.7
# Washington        4.0     145       73 26.2
```

```
USArrests[USArrests$Rape >= 20 & USArrests$Rape <= 40,]

#                Murder Assault UrbanPop Rape
# Alabama          13.2     236       58 21.2
# Arizona           8.1     294       80 31.0
# Colorado          7.9     204       78 38.7
# Florida          15.4     335       80 31.9
# Georgia          17.4     211       60 25.8
# Hawaii            5.3      46       83 20.2
# Illinois         10.4     249       83 24.0
# Indiana           7.2     113       65 21.0
# Louisiana        15.4     249       66 22.2
# Maryland         11.3     300       67 27.8
# Michigan         12.1     255       74 35.1
# Missouri          9.0     178       70 28.2
# New Mexico       11.4     285       70 32.1
# New York         11.1     254       86 26.1
# Ohio              7.3     120       75 21.4
# Oklahoma          6.6     151       68 20.0
# Oregon            4.9     159       67 29.3
# South Carolina   14.4     279       48 22.5
# Tennessee        13.2     188       59 26.9
# Texas            12.7     201       80 25.5
# Utah              3.2     120       80 22.9
# Virginia          8.5     156       63 20.7
# Washington        4.0     145       73 26.2
```

(c)
```
subset(USArrests, Murder >= 10 & Rape >= 20 & Rape <= 40)

#                Murder Assault UrbanPop Rape
# Alabama          13.2     236       58 21.2
# Florida          15.4     335       80 31.9
# Georgia          17.4     211       60 25.8
# Illinois         10.4     249       83 24.0
# Louisiana        15.4     249       66 22.2
# Maryland         11.3     300       67 27.8
# Michigan         12.1     255       74 35.1
# New Mexico       11.4     285       70 32.1
# New York         11.1     254       86 26.1
# South Carolina   14.4     279       48 22.5
# Tennessee        13.2     188       59 26.9
# Texas            12.7     201       80 25.5
```

(d) This time, we have to get a vector. We can create a data frame with the subset of data and directly subset it again (which looks a bit odd). Or we can use the `drop` argument from `subset`. Setting it to `TRUE` removes the surrounding data frame if a single column is selected.

```
subset(USArrests, Assault > 200)$UrbanPop

#  [1] 58 48 80 91 78 72 80 60 83 66 67 74 44 81 70
# [16] 86 45 48 80

subset(USArrests, Assault > 200, select = UrbanPop, drop = TRUE)

#  [1] 58 48 80 91 78 72 80 60 83 66 67 74 44 81 70
# [16] 86 45 48 80

USArrests[USArrests$Assault > 200,][[3]] # UrbanPop is the third column.

#  [1] 58 48 80 91 78 72 80 60 83 66 67 74 44 81 70
# [16] 86 45 48 80
```

Another very nice way is to construct an environment, in which we can refer to the column names directly (we will see in Part VII how this exactly works).

```
with(USArrests, UrbanPop[Assault > 200])

#  [1] 58 48 80 91 78 72 80 60 83 66 67 74 44 81 70
# [16] 86 45 48 80
```

Bonus exercise: As explained the `order` function returns positions that arranges its argument in ascending order.

```
order(USArrests$Assault)

#  [1] 34 11 45 49 15 29 23 48 19 41 27 38 17 26  7
# [16] 14 16 12 35 44 47 21 36 46 30 37 50 39 25 42
# [31]  4 43  6 10  1  8 13 18 28 32 22 24  2  5 40
# [46] 31  3 20  9 33
```

This basically means that the 34. element in the Assault column is the smallest value, the 11. element is the second smallest and so on. So we can take these values for reordering the whole data frame. Please note, that this creates an ordered copy of the original data frame. It does not order the existing data frame in place!

```
USArrests[order(USArrests$Assault),]

#                Murder Assault UrbanPop Rape
# North Dakota      0.8      45       44  7.3
# Hawaii            5.3      46       83 20.2
```

```
# Vermont          2.2      48        32 11.2
# Wisconsin        2.6      53        66 10.8
# Iowa             2.2      56        57 11.3
# New Hampshire    2.1      57        56  9.5
# Minnesota        2.7      72        66 14.9
# West Virginia    5.7      81        39  9.3
# Maine            2.1      83        51  7.8
# South Dakota     3.8      86        45 12.8
# Nebraska         4.3     102        62 16.5
# Pennsylvania     6.3     106        72 14.9
# Kentucky         9.7     109        52 16.3
# Montana          6.0     109        53 16.4
# Connecticut      3.3     110        77 11.1
# Indiana          7.2     113        65 21.0
# Kansas           6.0     115        66 18.0
# Idaho            2.6     120        54 14.2
# Ohio             7.3     120        75 21.4
# Utah             3.2     120        80 22.9
# Washington       4.0     145        73 26.2
# Massachusetts    4.4     149        85 16.3
# Oklahoma         6.6     151        68 20.0
# Virginia         8.5     156        63 20.7
# New Jersey       7.4     159        89 18.8
# Oregon           4.9     159        67 29.3
# Wyoming          6.8     161        60 15.6
# Rhode Island     3.4     174        87  8.3
# Missouri         9.0     178        70 28.2
# Tennessee       13.2     188        59 26.9
# Arkansas         8.8     190        50 19.5
# Texas           12.7     201        80 25.5
# Colorado         7.9     204        78 38.7
# Georgia         17.4     211        60 25.8
# Alabama         13.2     236        58 21.2
# Delaware         5.9     238        72 15.8
# Illinois        10.4     249        83 24.0
# Louisiana       15.4     249        66 22.2
# Nevada          12.2     252        81 46.0
# New York        11.1     254        86 26.1
# Michigan        12.1     255        74 35.1
# Mississippi     16.1     259        44 17.1
# Alaska          10.0     263        48 44.5
# California       9.0     276        91 40.6
# South Carolina  14.4     279        48 22.5
# New Mexico      11.4     285        70 32.1
# Arizona          8.1     294        80 31.0
```

```
# Maryland          11.3     300        67 27.8
# Florida           15.4     335        80 31.9
# North Carolina    13.0     337        45 16.1
```