Introduction to Statistics and R

Working with data sets

Eric Stemmler

Khovd University

24.02.2021

1. Recap: Exercise Solutions
2. R Tutorial
3. Formatting a data set
4. R - reading in data
5. Exercises
6. Quiz

Section 1

Recap: Exercise Solutions

Circles

1. Calculate the area of a circle $A = 2\pi r^2$ with $r = 2$ (Qian, 2016)
2. Write the circle area formula as a function with named as `circle` with parameter `r` and and calculate the area for `r <- seq(0, 3, 0.1)`
3. **Extension:** Extend your function to return a named `list` that contains the area and the circumference for a given parameter vector `r`

Normal Probability Density Function

1. Calculate the density of the normal probability distribution function $x \sim \mathcal{N}(2, 1.25)$ (mean $\mu = 2$ and standard deviation $\sigma = 1.25$) at `x <- seq(0, 4, 0.5)` by using the normal probability density formula $\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right)$, and verify your result by using the function `dnorm` (Qian, 2016).

Subsection 1

Solutions

Circles

```
circle <- function(r) {
    result <- list(area = 2 * pi * r^2,
                   circumference = 2 * pi * r)
    return(result)
}

circle(r = seq(0, 3, 0.5))
```

```
## $area
## [1]  0.000000  1.570796  6.283185 14.137167 25.132741
## [6] 39.269908 56.548668
##
## $circumference
## [1]  0.000000  3.141593  6.283185  9.424778 12.566371
## [6] 15.707963 18.849556
```

Normal Probability Density Function

```
npdf <- function(x, avg, stdev) {
    result <- 1.0 / (sqrt(2*pi*stdev^2)) *
        exp(-(x - avg)^2 / (2.0 * stdev^2))
    return(result)
}

npdf(x = seq(0, 4, 0.5), avg = 2, stdev = 1.25)
```

```
## [1] 0.08873667 0.15534884 0.23175324 0.29461611
## [5] 0.31915382 0.29461611 0.23175324 0.15534884
## [9] 0.08873667
```

```
    dnorm(x = seq(0, 4, 0.5), mean = 2, sd = 1.25)
```

```
## [1] 0.08873667 0.15534884 0.23175324 0.29461611
## [5] 0.31915382 0.29461611 0.23175324 0.15534884
## [9] 0.08873667
```

Section 2

R Tutorial

R Tutorial

- Every Thursday, 02:00-03:00pm, Room: 314 (here)
- Practising R and working on exercises
- No lecture, just programming practise and answering questions
- Also possible: analysis of your data sets

Section 3

Formatting a data set

CSV files

- CSV - **c**omma **s**eparated **v**alues
- easiest data format to work with
- can also be used by Microsoft Excel, etc.
- can easily be created from Microsoft Excel file (.xlsx)
- is not a Microsoft Excel Spreadsheet!

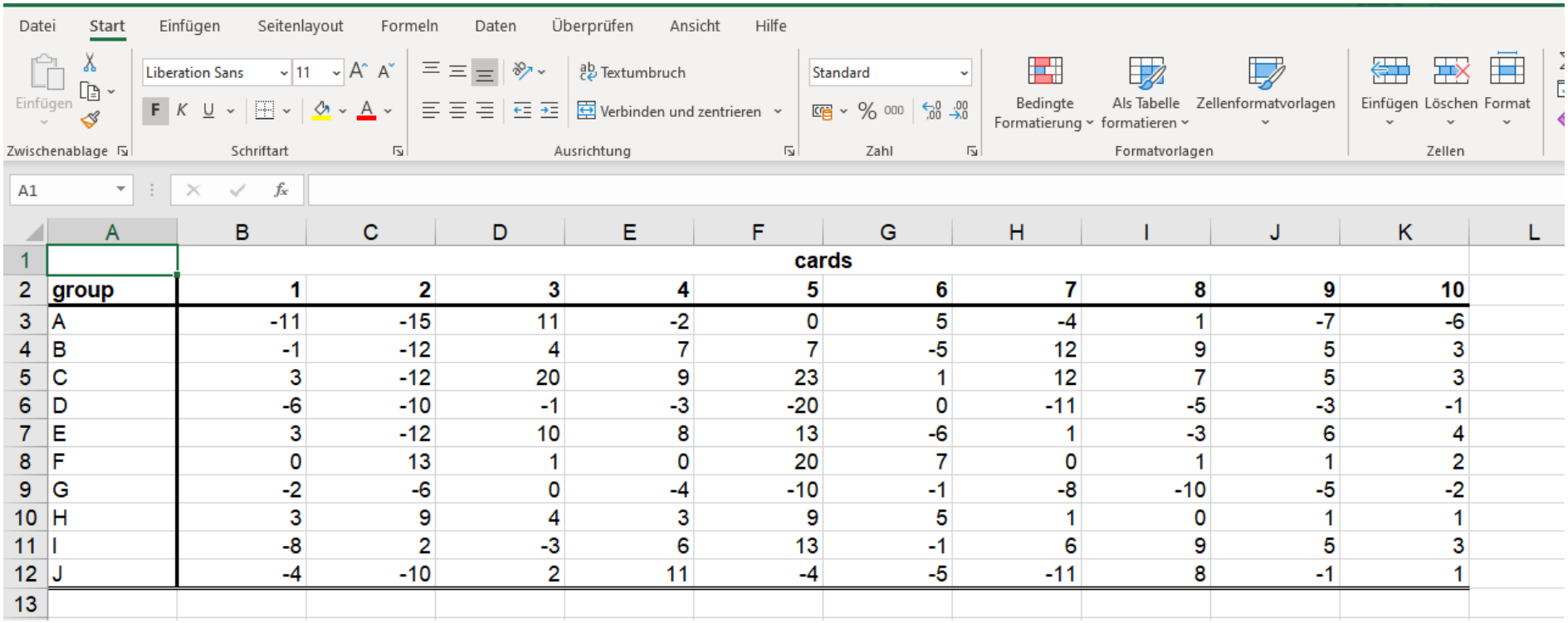
CSV files

Example: `test.csv`

```
name;age;language;nationality;
eric;33;english;german;
stefanie;32;russian;german;
nino;2;mongolian;german;
```

- Open any text editor
- Enter example text
- File » Save As "test.csv"
- Open "test.csv" in Microsoft Excel

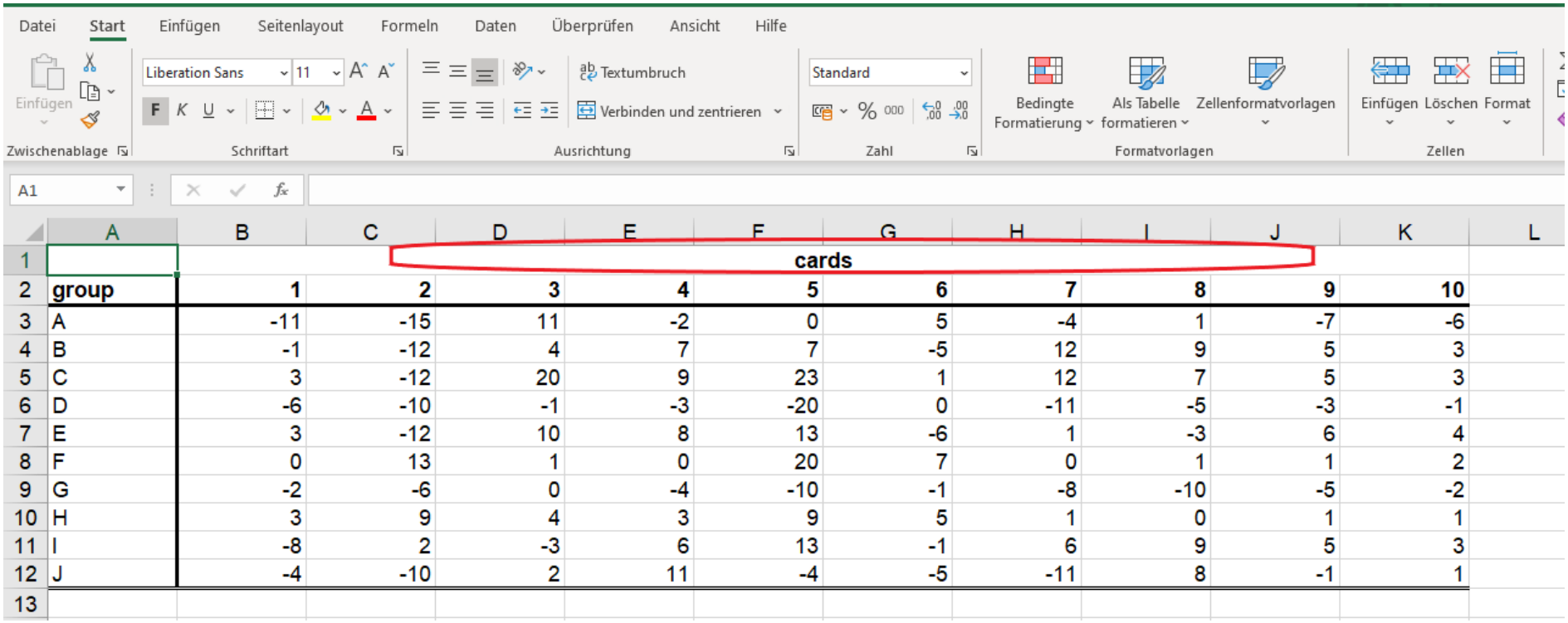
From Excel-file to CSV-file



	A	B	C	D	E	F	G	H	I	J	K	L
1						cards						
2	**group**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
3	A	-11	-15	11	-2	0	5	-4	1	-7	-6	
4	B	-1	-12	4	7	7	-5	12	9	5	3	
5	C	3	-12	20	9	23	1	12	7	5	3	
6	D	-6	-10	-1	-3	-20	0	-11	-5	-3	-1	
7	E	3	-12	10	8	13	-6	1	-3	6	4	
8	F	0	13	1	0	20	7	0	1	1	2	
9	G	-2	-6	0	-4	-10	-1	-8	-10	-5	-2	
10	H	3	9	4	3	9	5	1	0	1	1	
11	I	-8	2	-3	6	13	-1	6	9	5	3	
12	J	-4	-10	2	11	-4	-5	-11	8	-1	1	
13												

Typical Microsoft Excel table. Not suitable for using with R.

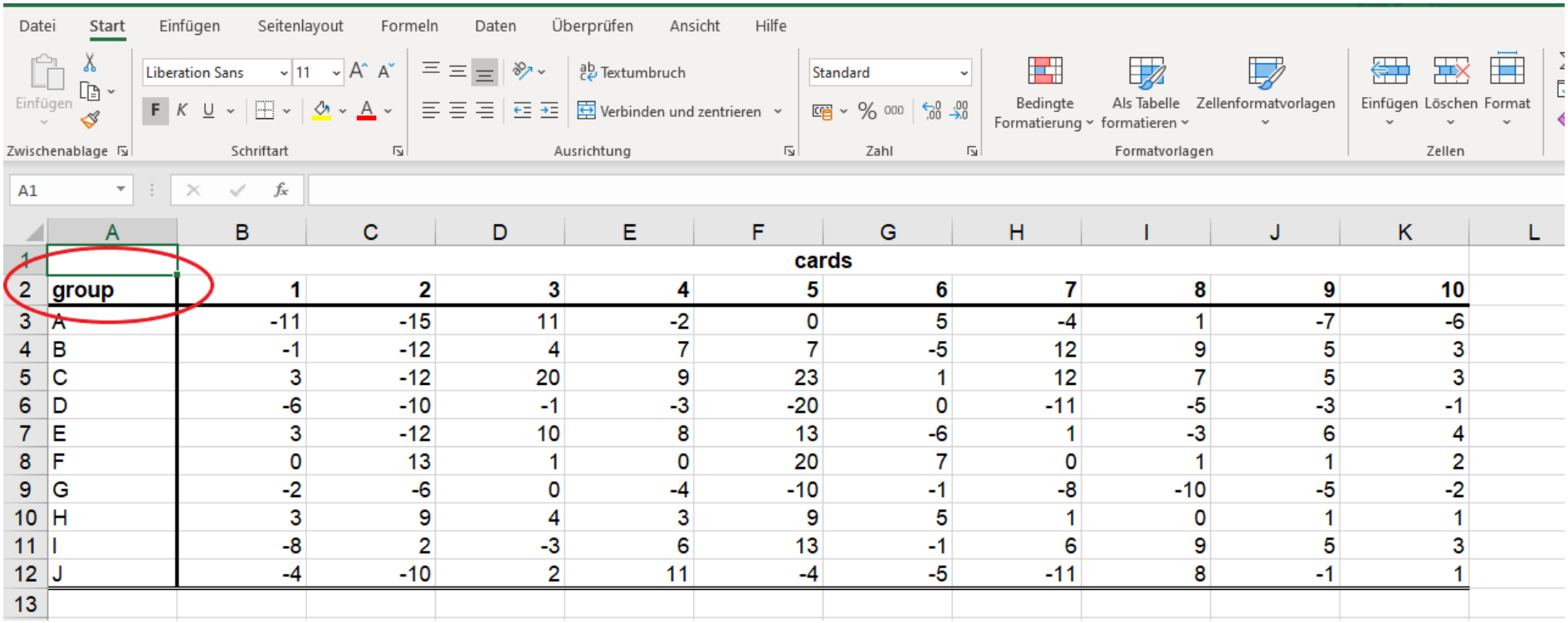
From Excel-file to CSV-file



	A	B	C	D	E	F	G	H	I	J	K	L
1						cards						
2	**group**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
3	A	-11	-15	11	-2	0	5	-4	1	-7	-6	
4	B	-1	-12	4	7	7	-5	12	9	5	3	
5	C	3	-12	20	9	23	1	12	7	5	3	
6	D	-6	-10	-1	-3	-20	0	-11	-5	-3	-1	
7	E	3	-12	10	8	13	-6	1	-3	6	4	
8	F	0	13	1	0	20	7	0	1	1	2	
9	G	-2	-6	0	-4	-10	-1	-8	-10	-5	-2	
10	H	3	9	4	3	9	5	1	0	1	1	
11	I	-8	2	-3	6	13	-1	6	9	5	3	
12	J	-4	-10	2	11	-4	-5	-11	8	-1	1	
13												

Cells should not be merged.

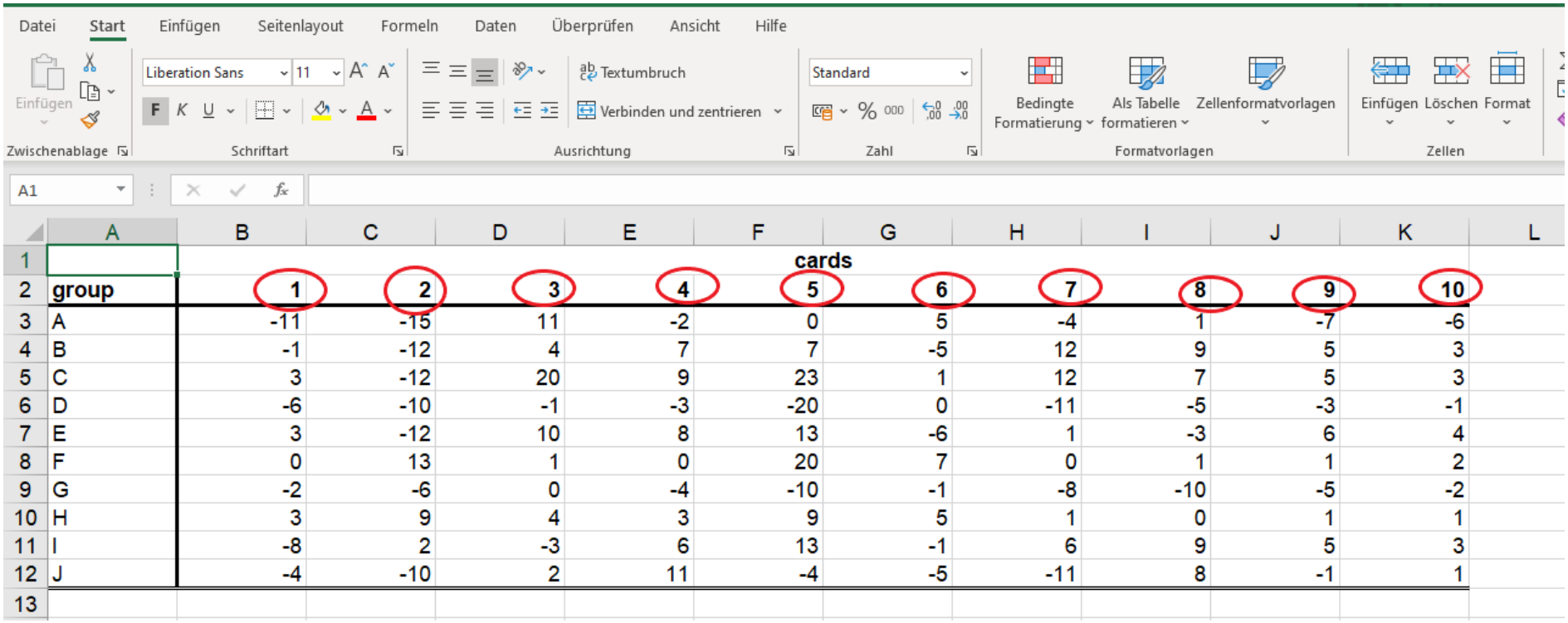
From Excel-file to CSV-file



	A	B	C	D	E	F	G	H	I	J	K	L
1						cards						
2	**group**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
3	A	-11	-15	11	-2	0	5	-4	1	-7	-6	
4	B	-1	-12	4	7	7	-5	12	9	5	3	
5	C	3	-12	20	9	23	1	12	7	5	3	
6	D	-6	-10	-1	-3	-20	0	-11	-5	-3	-1	
7	E	3	-12	10	8	13	-6	1	-3	6	4	
8	F	0	13	1	0	20	7	0	1	1	2	
9	G	-2	-6	0	-4	-10	-1	-8	-10	-5	-2	
10	H	3	9	4	3	9	5	1	0	1	1	
11	I	-8	2	-3	6	13	-1	6	9	5	3	
12	J	-4	-10	2	11	-4	-5	-11	8	-1	1	
13												

The first row should be used for table header/ variable names

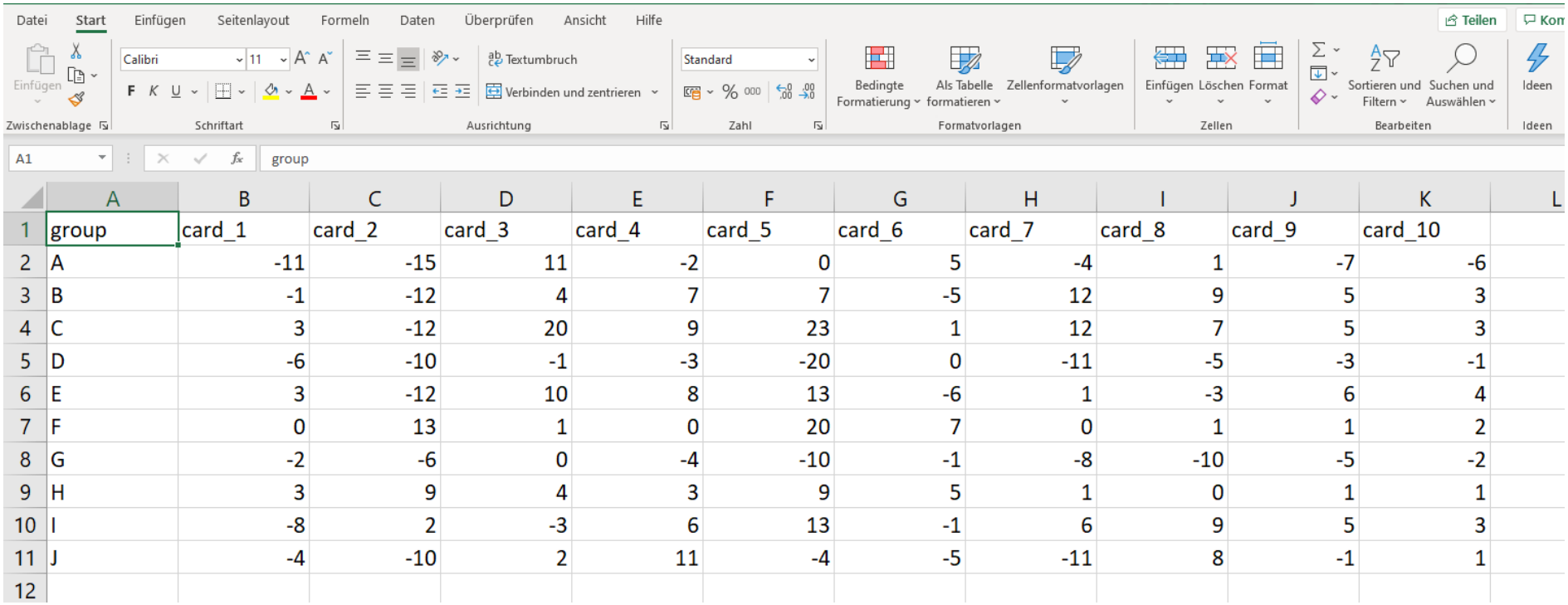
From Excel-file to CSV-file



	A	B	C	D	E	F	G	H	I	J	K	L
1						cards						
2	group	1	2	3	4	5	6	7	8	9	10	
3	A	-11	-15	11	-2	0	5	-4	1	-7	-6	
4	B	-1	-12	4	7	7	-5	12	9	5	3	
5	C	3	-12	20	9	23	1	12	7	5	3	
6	D	-6	-10	-1	-3	-20	0	-11	-5	-3	-1	
7	E	3	-12	10	8	13	-6	1	-3	6	4	
8	F	0	13	1	0	20	7	0	1	1	2	
9	G	-2	-6	0	-4	-10	-1	-8	-10	-5	-2	
10	H	3	9	4	3	9	5	1	0	1	1	
11	I	-8	2	-3	6	13	-1	6	9	5	3	
12	J	-4	-10	2	11	-4	-5	-11	8	-1	1	
13												

- Variable names should be not start with a number or other non-readable characters.
- Variable names cannot contain characters like £, €, $, %, ^, *, +, −, (,), [ ], #, !, ?, <, > or anything that is a operator in R

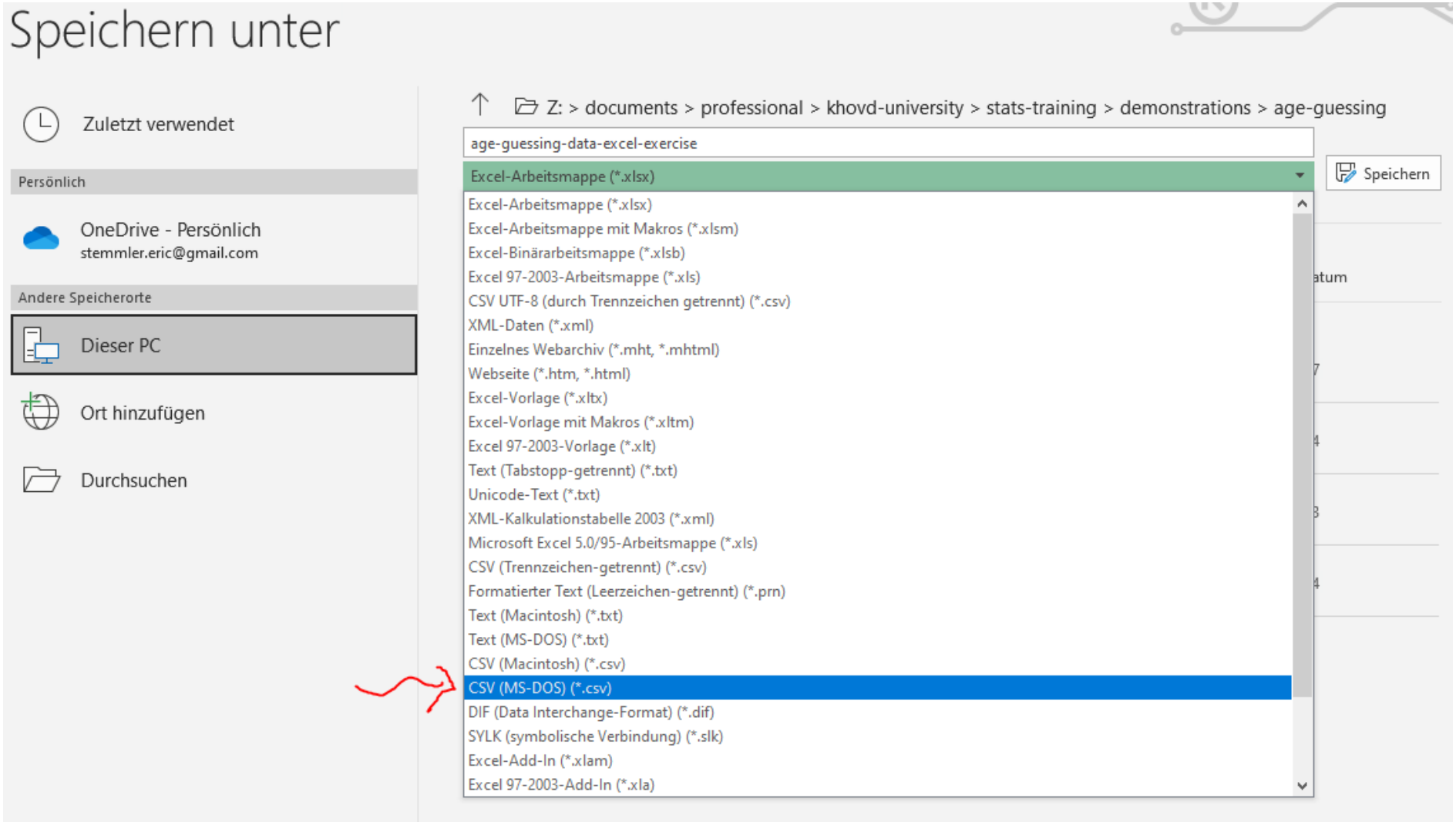
From Excel-file to CSV-file



	A	B	C	D	E	F	G	H	I	J	K	L
1	group	card_1	card_2	card_3	card_4	card_5	card_6	card_7	card_8	card_9	card_10	
2	A	-11	-15	11	-2	0	5	-4	1	-7	-6	
3	B	-1	-12	4	7	7	-5	12	9	5	3	
4	C	3	-12	20	9	23	1	12	7	5	3	
5	D	-6	-10	-1	-3	-20	0	-11	-5	-3	-1	
6	E	3	-12	10	8	13	-6	1	-3	6	4	
7	F	0	13	1	0	20	7	0	1	1	2	
8	G	-2	-6	0	-4	-10	-1	-8	-10	-5	-2	
9	H	3	9	4	3	9	5	1	0	1	1	
10	I	-8	2	-3	6	13	-1	6	9	5	3	
11	J	-4	-10	2	11	-4	-5	-11	8	-1	1	
12												

Re-formated Microsoft Excel table suitable for using with R. Needs to be saved as `*.csv` file.

From Excel-file to CSV-file



Section 4

R - reading in data

R - reading in data

```
# On windows operating system
read.csv(file = "C:/Users/eric/Desktop/age-guessing-data-excel-exercise.csv")
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6 card_7 card_8 card_9 card_10
## 1      A    -11    -15     11     -2      0      5     -4      1     -7     -6
## 2      B     -1    -12      4      7      7     -5     12      9      5      3
## 3      C      3    -12     20      9     23      1     12      7      5      3
## 4      D     -6    -10     -1     -3    -20      0    -11     -5     -3     -1
## 5      E      3    -12     10      8     13     -6      1     -3      6      4
## 6      F      0     13      1      0     20      7      0      1      1      2
## 7      G     -2     -6      0     -4    -10     -1     -8    -10     -5     -2
## 8      H      3      9      4      3      9      5      1      0      1      1
## 9      I     -8      2     -3      6     13     -1      6      9      5      3
## 10     J     -4    -10      2     11     -4     -5    -11      8     -1      1
```

Ctrl + R (script editor)

Enter (console)

R - reading in data

```
# install.packages("data.table")
library(data.table)
dt <- fread(input = "age-guessing-data-excel-exercise.csv")
dt
```

R - reading in data

```
##     group card_1 card_2 card_3 card_4 card_5 card_6 card_7 card_8 card_9
##  1:     A    -11    -15     11     -2      0      5     -4      1     -7
##  2:     B     -1    -12      4      7      7     -5     12      9      5
##  3:     C      3    -12     20      9     23      1     12      7      5
##  4:     D     -6    -10     -1     -3    -20      0    -11     -5     -3
##  5:     E      3    -12     10      8     13     -6      1     -3      6
##  6:     F      0     13      1      0     20      7      0      1      1
##  7:     G     -2     -6      0     -4    -10     -1     -8    -10     -5
##  8:     H      3      9      4      3      9      5      1      0      1
##  9:     I     -8      2     -3      6     13     -1      6      9      5
## 10:     J     -4    -10      2     11     -4     -5    -11      8     -1
##     card_10
##  1:      -6
##  2:       3
##  3:       3
##  4:      -1
##  5:       4
##  6:       2
##  7:      -2
##  8:       1
##  9:       3
## 10:       1
```

R - reading in data

```
dt[, total_error := abs(card_1) +
   abs(card_2) +
   abs(card_3) +
   abs(card_4) +
   abs(card_5) +
   abs(card_6) +
   abs(card_7) +
   abs(card_8) +
   abs(card_9) +
   abs(card_10)]
dt
```

- Function `abs(x)`: Compute absolute value of x
- Operator `:=`: Create new column in data table

R - reading in data

```
##     group card_1 card_2 card_3 card_4 card_5 card_6 card_7 card_8 card_9
##  1:     A    -11    -15     11     -2      0      5     -4      1     -7
##  2:     B     -1    -12      4      7      7     -5     12      9      5
##  3:     C      3    -12     20      9     23      1     12      7      5
##  4:     D     -6    -10     -1     -3    -20      0    -11     -5     -3
##  5:     E      3    -12     10      8     13     -6      1     -3      6
##  6:     F      0     13      1      0     20      7      0      1      1
##  7:     G     -2     -6      0     -4    -10     -1     -8    -10     -5
##  8:     H      3      9      4      3      9      5      1      0      1
##  9:     I     -8      2     -3      6     13     -1      6      9      5
## 10:     J     -4    -10      2     11     -4     -5    -11      8     -1
##     card_10 total_error
##  1:      -6          62
##  2:       3          65
##  3:       3          95
##  4:      -1          60
##  5:       4          66
##  6:       2          45
##  7:      -2          48
##  8:       1          36
##  9:       3          56
## 10:       1          57
```

R - reading in data

```
dt[, mean_error := total_error / 10]
head(dt)
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6 card_7 card_8 card_9 card_10
## 1:     A    -11    -15     11     -2      0      5     -4      1     -7     -6
## 2:     B     -1    -12      4      7      7     -5     12      9      5      3
## 3:     C      3    -12     20      9     23      1     12      7      5      3
## 4:     D     -6    -10     -1     -3    -20      0    -11     -5     -3     -1
## 5:     E      3    -12     10      8     13     -6      1     -3      6      4
## 6:     F      0     13      1      0     20      7      0      1      1      2
##    total_error mean_error
## 1:          62        6.2
## 2:          65        6.5
## 3:          95        9.5
## 4:          60        6.0
## 5:          66        6.6
## 6:          45        4.5
```

- Function `head(x)`: Print the first 6 rows of x

R - reading in data

```
fwrite(x = dt, filename = "my-data-set.csv", sep = ";")
```

- Function `fwrite(x)`: Save data.table `x` as `*.csv` file
- ... as file named `filename`
- ... using separating character `sep`

Section 5

Exercises

Subsection 1

Creating data sets

c()-function

Epidemiological data set: Observations of red deer and wild boar, tubercolosis (tb), parasite (ecervi) (Zuur et al., 2009)

farm	month	year	sex	length.class	length.ct	ecervi	tb
MO	11	2000	1	1	75.0	0	0
MO	7	2000	2	1	85.0	0	0
MO	7	2001	2	1	91.6	0	1
MO	NA	NA	2	1	95.0	NA	NA
LN	9	2003	1	1	NA	0	0
SE	9	2003	2	1	105.5	0	0
QM	11	2002	2	1	106.0	0	0

1. Use `c()` to create a vector of the `tb` values. Store the result in a variable named `tb`.

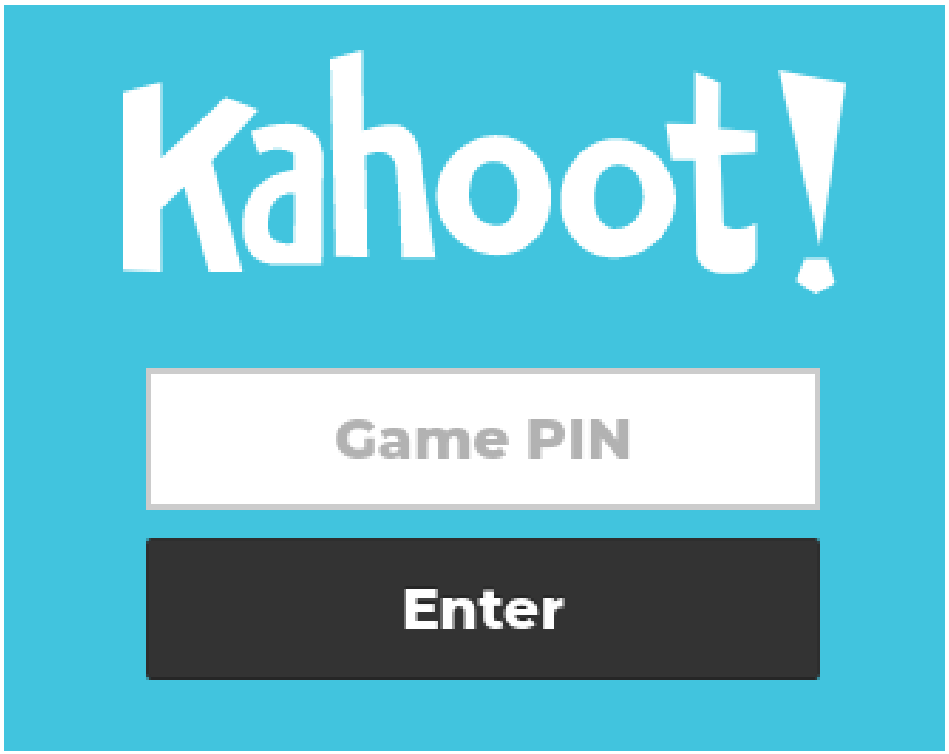
list()-function

2. Use `c()` to create a second vector of the `length.ct` values. Store the result in a variable named `length.ct`.
3. Use `list()` to create a list using the two vectors `length.ct` and `tb`. Store the result as `deer`
4. Use the function `as.data.table()` to turn `deer` into a data.table. Print the result to the console.

Section 6

Quiz

Quiz



- Phone or Computer: https://www.kahoot.it
- Wifi: Platinum
- Password: H2SvsH2O

Song S Qian. *Environmental and ecological statistics with R*. CRC press, 2016.

Alain Zuur, Elena N Ieno, and Erik Meesters. *A Beginner's Guide to R*. Springer Science & Business Media, 2009.