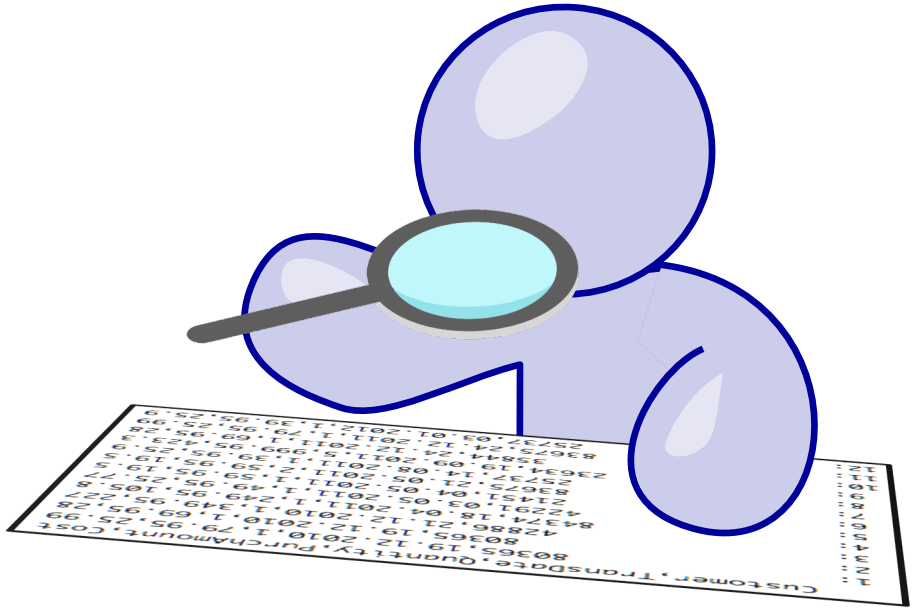
Basic techniques for investigating data objects

Observe and explore your data:
3 options to make sure the data is loaded correctly

Many mistakes can be made when loading data. Checking the data before working with it is always a good idea:

1. Look at the data
2. Look at the individual variables
3. Look at summary statistics



Step 1:
Look at your data

	Customer	TransDate	Quantity	PurchAmount	Cost	TransID
1	149332	15.11.2005	1	199.95	107.00	27998739
2	172951	29.08.2008	1	199.95	108.00	128888288
3	120621	19.10.2007	1	99.95	49.00	125375247
4	149236	14.11.2005	1	39.95	18.95	127996226
5	149236	12.06.2007	1	79.95	35.00	128670302
...	...	...	...	...	...	...
223187	199997	17.09.2012	1	29.95	13.80	132481149
223188	199997	17.09.2012	1	29.95	13.80	132481149
223189	199998	17.09.2012	1	29.95	13.80	132481154
223190	199999	17.09.2012	1	179.95	109.99	132481165
223191	199542	17.09.2012	1	39.95	10.50	131973368

[223191 rows x 5 columns]

myData

Step 1:
Look at your data

Look at the first observations with the `head()` function:

```
head(myData, n=3)
```

```
  Customer  TransDate  Quantity PurchAmount   Cost   TransID
1   149332 15.11.2005        1      199.95 107.00  27998739
2   172951 29.08.2008        1      199.95 108.00 128888288
3   120621 19.10.2007        1       99.95  49.00 125375247
```

Do the same for the last observations with the `tail()` function:

```
tail(myData, n=3)
```

```
       Customer    TransDate  Quantity PurchAmount   Cost   TransID
223189   199998 17.09.2012        1       29.95  13.80 132481154
223190   199999 17.09.2012        1      179.95 109.99 132481165
223191   199542 17.09.2012        1       39.95  10.50 131973368
```

Step 1:
Look at your data

Look at the first observations with the `head()` function:

```
head(myData, n=3)
```

```
  Customer  TransDate  Quantity PurchAmount   Cost   TransID
1   149332 15.11.2005        1      199.95 107.00  27998739
2   172951 29.08.2008        1      199.95 108.00 128888288
3   120621 19.10.2007        1       99.95  49.00 125375247
```

Do the same for the last observations with the `tail()` function:

```
tail(myData, n=3)
```

```
       Customer    TransDate  Quantity PurchAmount   Cost   TransID
223189   199998 17.09.2012        1       29.95  13.80 132481154
223190   199999 17.09.2012        1      179.95 109.99 132481165
223191   199542 17.09.2012        1       39.95  10.50 131973368
```

Step 1:
Look at your data

Look at the first observations with the `head()` function:

```
head(myData, n=3)
```

```
  Customer  TransDate  Quantity PurchAmount   Cost   TransID
1   149332 15.11.2005        1      199.95 107.00  27998739
2   172951 29.08.2008        1      199.95 108.00 128888288
3   120621 19.10.2007        1       99.95  49.00 125375247
```

Do the same for the last observations with the `tail()` function:

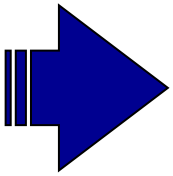
```
tail(myData, n=3)
```

```
       Customer    TransDate  Quantity PurchAmount   Cost   TransID
223189   199998 17.09.2012        1       29.95  13.80 132481154
223190   199999 17.09.2012        1      179.95 109.99 132481165
223191   199542 17.09.2012        1       39.95  10.50 131973368
```

Step 2:
Look at individual variables

Before processing any data, you always have to ensure that your data is formatted properly and that the right data types are assigned to your variables. This will save a lot of time and you can avoid common mistakes.

Customer	TransDate	Quantity	PurchAmount	Cost	TransID
149332	15/11/05	1	199.95	107.00	127998739
172951	29/08/08	1	199.95	108.00	128888288
120621	19/10/07	1	99.95	49.00	125375247
149236	14/11/05	1	39.95	18.95	127996226
149236	12/06/07	1	79.95	35.00	128670302
…	…	…	…	…	…



```
Classes 'data.table' and 'data.frame':  223191 c
 $ Customer   : int  149332 172951 120621 149236
 $ TransDate  : chr  "15/11/05" "29/08/08" "19/1
 $ Quantity   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ PurchAmount: num  200 200 100 40 80 ...
 $ Cost       : num  107 108 49 18.9 35 ...
 $ TransID    : int  127998739 128888288 125375
 - attr(*, ".internal.selfref")=<externalptr>
```

Check if the type of the variables is correct.

```
str(myData)
```

Sidenote: Built-in data types in R

R distinguishes between several data types. The most common are:

Data type		Description	Sign	Example
Logical		Variable is a logical value which can either be *True* or *False.*	bool	*True, False*
Numeric	integer/ float	Variable is a number which can be written without a fractional component (whole-number) or a computational approximation of any real-valued number.	num	*-3, 0, 1, 2, 3,…* *-2.6, 1.0, 1.1, 1.329*
Character	string	Variable is interpreted as "text".	char	*"a", "Z", "Hello", "Anna"*
Factor		Variable is a factor (several levels).	factor w/ xxx levels	*factor(c("brown", "yellow", "green"))*
Dates and time	datetime	Variable is a data or time and special functionalities for manipulation are provided.	POSIXct	*d=date(2005, 7, 14)* *t=time(12, 30)* *datetime.combine(d, t)*

- (Usually) R automatically selects the right data format (except for dates).

Sidenote: Built-in data types in R

R distinguishes between several data types. The most common are:

Data type		Description	Sign	Example
Logical		Variable is a logical value which can either be *True* or *False.*	bool	*True, False*
Numeric	integer/ float	Variable is a number which can be written without a fractional component (whole-number) or a computational approximation of any real-valued number.	num	*-3, 0, 1, 2, 3,…* *-2.6, 1.0, 1.1, 1.329*
Character	string	Variable is interpreted as "text".	char	*"a", "Z", "Hello", "Anna"*
Factor		Variable is a factor (several levels).	factor w/ xxx levels	*factor(c("London", "Berlin", "Paris"))*
Dates and time	datetime	Variable is a data or time and special functionalities for manipulation are provided.	POSIXct	*d=date(2005, 7, 14)* *t=time(12, 30)* *datetime.combine(d, t)*

- (Usually) R automatically selects the right data format (except for dates).

Sidenote: Built-in data types in R

R distinguishes between several data types. The most common are:

Data type		Description	Sign	Example
Logical		Variable is a logical value which can either be *True* or *False.*	bool	*True, False*
Numeric	integer/ float	Variable is a number which can be written without a fractional component (whole-number) or a computational approximation of any real-valued number.	num	*-3, 0, 1, 2, 3,…* *-2.6, 1.0, 1.1, 1.329*
Character	string	Variable is interpreted as "text".	char	*"a", "Z", "Hello", "Anna"*
Factor		Variable is a factor (several levels).	factor w/ xxx levels	*factor(c("brown", "yellow", "green"))*
Dates and time	datetime	Variable is a data or time and special functionalities for manipulation are provided.	POSIXct	*d=date(2005, 7, 14)* *t=time(12, 30)* *datetime.combine(d, t)*

- (Usually) R automatically selects the right data format (except for dates).

Sidenote: Package "lubridate"

"**lubridate**" makes it easier to work with dates and times:

- Identify and parse time
- Extract and modify years, months, days, hours, …
- Perform accurate math with date-times

Format the data

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
...	...	...	...	...

1 Characetrs

Customer	TransDate	...
149332	2005-11-15	...
172951	2008-08-29	...
120621	2007-10-19	...
149236	2005-11-14	...
149236	2007-06-12	...
...	...	...

5 Recognized as date

2 data.table object to modify

3 Column to modify

```
myData[, TransDate:=dmy(TransDate, tz="UTC")]
```

4 Function is part of the `lubridate` package

Format the data

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
...	...	...	...	...

1 Characetrs

Customer	TransDate	...
149332	2005-11-15	...
172951	2008-08-29	...
120621	2007-10-19	...
149236	2005-11-14	...
149236	2007-06-12	...
...	...	...

5 Recognized as date

2 data.table object to modify

3 Column to modify

```
myData[, TransDate:=dmy(TransDate, tz="UTC")]
```

4 Function is part of the `lubridate` package

Format the data

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
...	...	...	...	...

1 Characetrs

Customer	TransDate	...
149332	2005-11-15	...
172951	2008-08-29	...
120621	2007-10-19	...
149236	2005-11-14	...
149236	2007-06-12	...
...	...	...

5 Recognized as date

2 data.table object to modify

3 Column to modify

```
myData[, TransDate:=dmy(TransDate, tz="UTC")]
```

4 Function is part of the `lubridate` package

Format the data

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
...	...	...	...	...

1 Characetrs

Customer	TransDate	...
149332	2005-11-15	...
172951	2008-08-29	...
120621	2007-10-19	...
149236	2005-11-14	...
149236	2007-06-12	...
...	...	...

5 Recognized as date

2 data.table object to modify

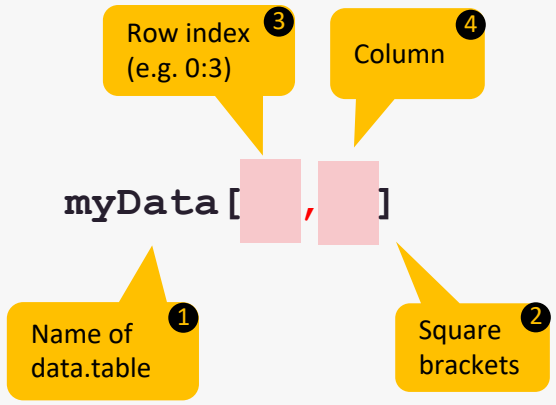
3 Column to modify

```
myData[, TransDate:=dmy(TransDate, tz="UTC")]
```

4 Function is part of the `lubridate` package

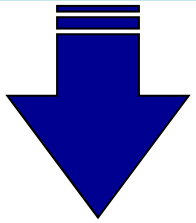
Sidenote: General command structure for data.table objects

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
...	...	...	...	...



Step 3:
Look at summary statistics

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15/11/05	1	199.95	107.00
172951	29/08/08	1	199.95	108.00
120621	19/10/07	1	99.95	49.00
149236	14/11/05	1	39.95	18.95
149236	12/06/07	1	79.95	35.00
…	…	…	…	…



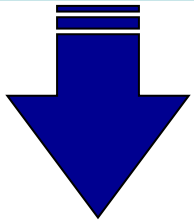
```
summary(myData)
```

Are the summary statistics as you expect them to be?

```
Customer            TransDate                Cost
length: 223191      Min. :    2004-12-16     Min. :    0.00
Class: character    1st Qu. : 2007-05-11     1st Qu.:  14.03
Mode: character     Median. : 2008-12-16     Median.:  24.00
                    Mean:     2009-01-12     Mean:     39.01
                    3rd Qu.:  2010-11-17     3rd Qu.:  45.00
                    Max:      2012-12-09     Max.:     3100.00
```

Step 3:
Look at summary statistics

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15/11/05	1	199.95	107.00
172951	29/08/08	1	199.95	108.00
120621	19/10/07	1	99.95	49.00
149236	14/11/05	1	39.95	18.95
149236	12/06/07	1	79.95	35.00
…	…	…	…	…



```
summary(myData)
```

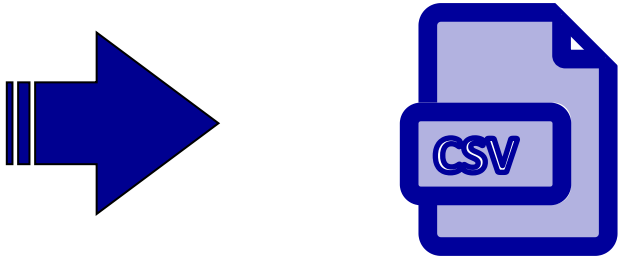
Are the summary statistics as you expect them to be?

```
Customer            TransDate
length: 223191      Min. :    2004-12-16
Class: character    1st Qu. : 2007-05-11
Mode: character     Median. : 2008-12-16
                    Mean:     2009-01-12
                    3rd Qu.:  2010-11-17
                    Max:      2012-12-09
```

```
Cost
Min. :     0.00
1st Qu.:   14.03
Median.:   24.00
Mean:      39.01
3rd Qu.:   45.00
Max.:      3100.00
```

Write data as CSV

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
…	…	…	…	…



R Base

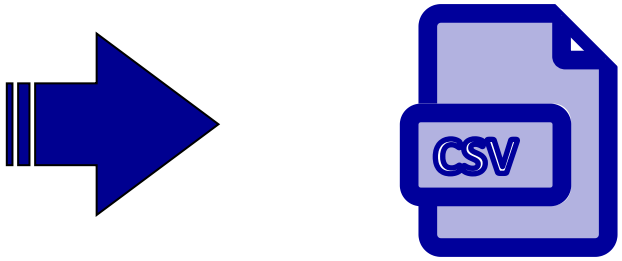
```
write.csv(myData,
          "myData.csv", …)
```

data.table

```
fwrite(myData,
       "myData.csv", ...)
```

Write data as CSV

Customer	TransDate	Quantity	PurchAmount	Cost
149332	15.11.2005	1	199.95	107.00
172951	29.08.2008	1	199.95	108.00
120621	19.10.2007	1	99.95	49.00
149236	14.11.2005	1	39.95	18.95
149236	12.06.2007	1	79.95	35.00
…	…	…	…	…



R Base

```
write.csv(myData,
          "myData.csv", …)
```

1. Object to save
2. Location and name of output CSV file

data.table

```
fwrite(myData,
       "myData.csv", ...)
```

Sidenote: Remove objects from your workspace

When you are finished with an object it is good practice (but not obligatory) to remove it from your workspace. Thus, you save storage and keep your programming environment clean:

Basic techniques for investigating data objects