Data Analytics in Business

Introduction to R Data Frames

MGT 6203 – Online – Fall 2020

Frederic Bien
fvbien@scheller.gatech.edu

Georgia Tech Scheller College of Business

Lessons

A. Data
B. Vectors and Data Frames

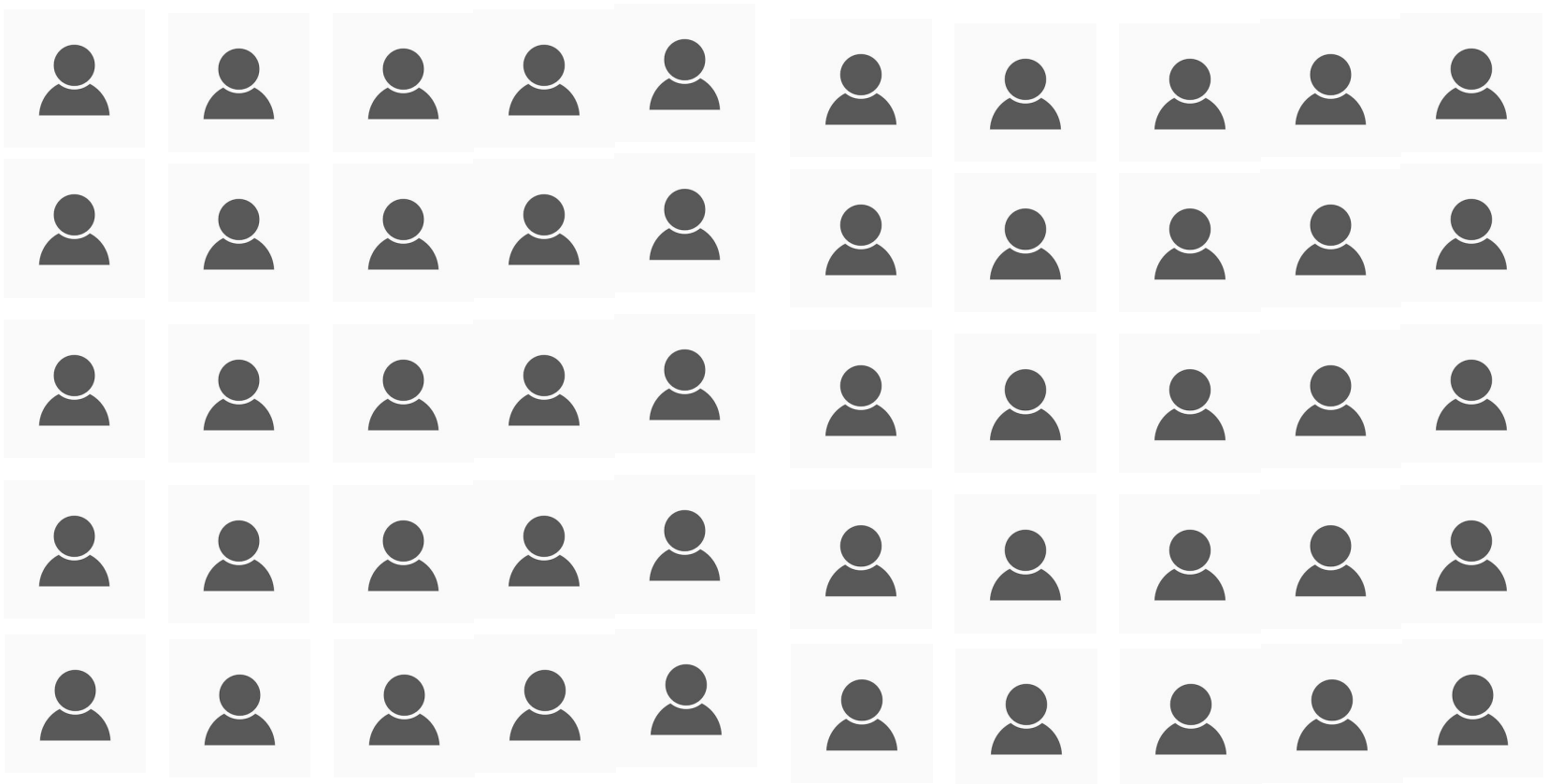
Reference: *https://cran.r-project.org/doc/manuals/r-release/R-intro.html*

A. Data

- Information / knowledge
- Measured / collected / reported.. -> for analysis!
- Measurement / collection error -> error in analysis

- Name: Peter Parker
- Gender: Male
- Age: 21
- College: Georgia Tech
- Major: Business
- Status: Junior
- Concentration: ITM
- GPA: 3.99
- Taken MGT 4050: YES
- Internships: 2
- Favorite food: pizza
- Job offers: 2
- …

Data

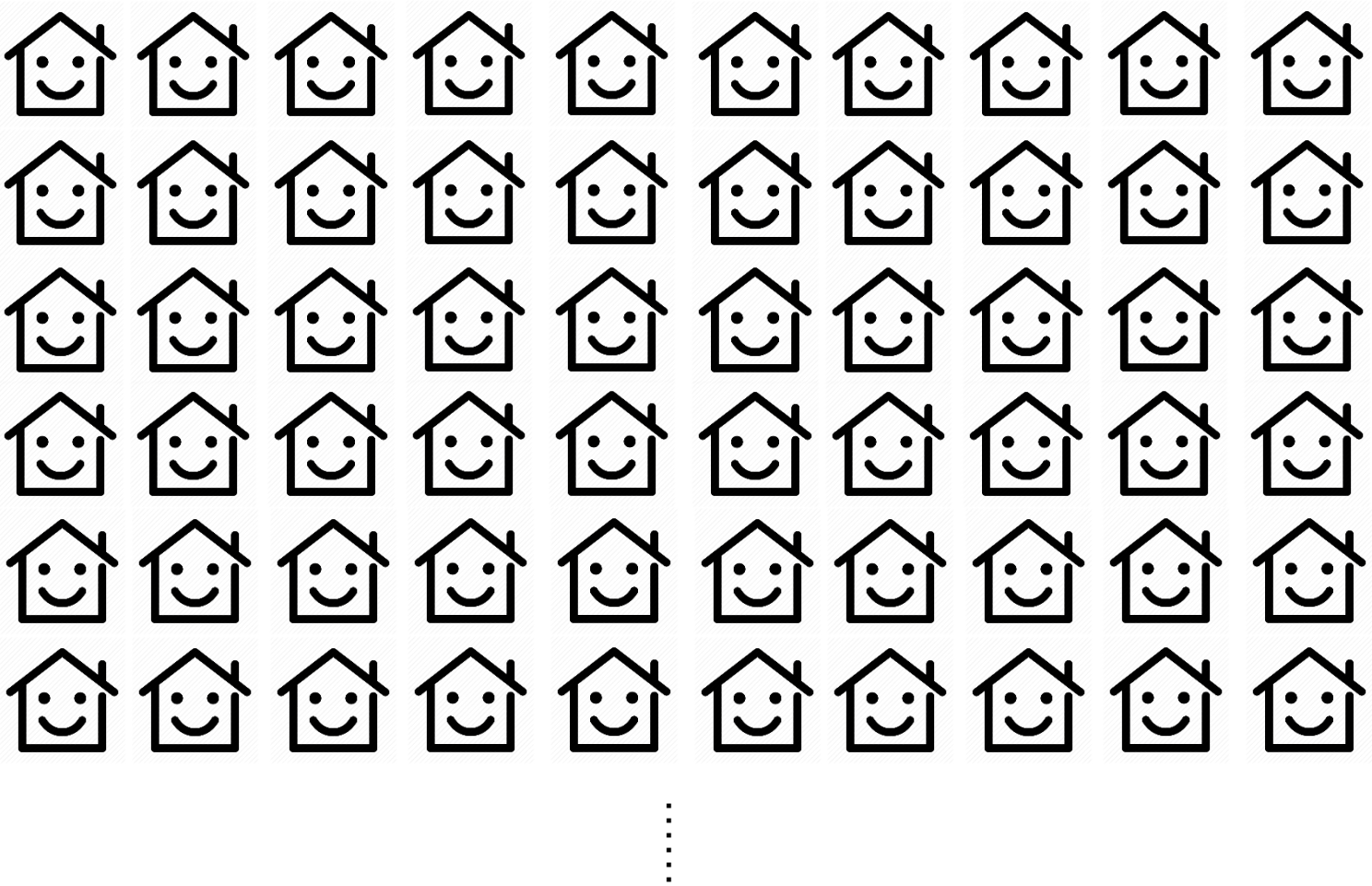


Data

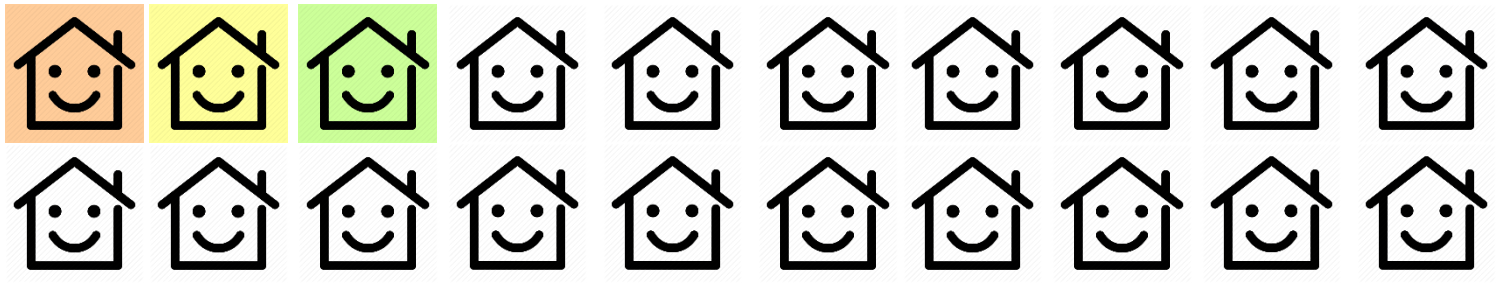
- Information / knowledge

- Sale price: 400K USD
- House size: 10K (sqft)
- Lot size: 20K (sqft)
- Bedrooms: 8
- Location: Long Beach, CA
- Sale date : 2019/03/10
- Built date: 2012/01/28
- Ocean view: no
- Rooftop : yes
- Color of the exterior wall : ivory
- Zip code: 12345
- # of previous owners : 2
- …

Data



Data



Price	Size	Lotsize	Bedrooms	Location	Sale date	Built date	Ocean view	Rooftop	Zipcode
400,000	10,000	20,000	8	Long Beach, CA	2019/03/10	2012/01/28	no	yes	12345
300,500	8,000	18,000	7	Boston, MA	2019/05/01	2013/02/26	no	no	31245
128,250	12,000	18,000	9	Phoenix, AZ	2019/03/08	2016/05/05	no	no	54321

Data

- *A set of values of **subjects** with respect to **qualitative / quantitative** variables (features)*

- *Subjects*
  - Could be anything
  - People – of certain characteristics, house, products, …

- *Qualitative variables*
  - Categorical
  - Not numerical

- *Quantitative variables*
  - Numerical

Data

Header***(column names)***

Price	Size	Lotsize	Bedrooms	Location	Sale date	Built date	Ocean view	Rooftop	Zipcode
400,000	10,000	20,000	8	Long Beach, CA	2019/03/10	2012/01/28	no	yes	12345
300,500	8,000	18,000	7	Boston, MA	2019/05/01	2013/02/26	no	no	31245
128,250	12,000	18,000	9	Phoenix, AZ	2019/03/08	2016/05/05	no	no	54321

- Data frames (data tables)

- *Rows*
- *Columns*

Data

Price	Size	Lotsize	Bedrooms	Location	Sale date	Built date	Ocean view	Rooftop	Zipcode
400,000	10,000	20,000	8	Long Beach, CA	2019/03/10	2012/01/28	no	yes	12345
300,500	8,000	18,000	7	Boston, MA	2019/05/01	2013/02/26	no	no	31245
128,250	12,000	18,000	9	Phoenix, AZ	2019/03/08	2016/05/05	no	no	54321

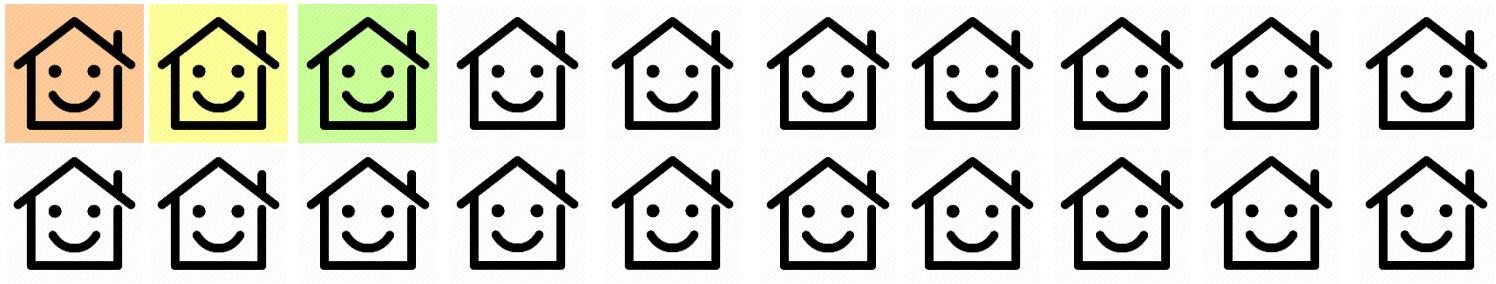
- Row ***(subjects)***
  - A single data item in a table
  - A set of related data
  - Every row in the table has the same structure
    - In a table that represents houses, each row represents a single house
    - i.e) data on a unit level – Yi = Y1, Y2, Y3.. (Y of the 1st, 2nd, 3rd observations)
    - Subjects – could reoccur with respect to time
    - i.e) data on a unit & datetime level – Yit (Y of the 1st on week 1, week 2..)

Data

Price	Size	Lotsize	Bedrooms	Location	Sale date	Built date	Ocean view	Rooftop	Zipcode
400,000	10,000	20,000	8	Long Beach, CA	2019/03/10	2012/01/28	no	yes	12345
300,500	8,000	18,000	7	Boston, MA	2019/05/01	2013/02/26	no	no	31245
128,250	12,000	18,000	9	Phoenix, AZ	2019/03/08	2016/05/05	no	no	54321

- Column ***(variables)***
  - Information / data about features of a subject
  - Each column has a **data value of a particular type**
  - A categorical column -> has categorical values
    - Logical
    - Character
  - A numerical column -> has numerical values
    - Numeric
    - Integer

Data



Price	Size	Lotsize	Bedrooms	Location	Sale date	Built date	Ocean view	Rooftop	Zipcode
400,000	10,000	20,000	8	Long Beach, CA	2019/03/10	2012/01/28	no	yes	12345
300,500	8,000	18,000	7	Boston, MA	2019/05/01	2013/02/26	no	no	31245
128,250	12,000	18,000	9	Phoenix, AZ	2019/03/08	2016/05/05	no	no	54321

Data

- Data Analysis
  - Examine related data sets
  - Understand: fit values into a statistical model (i.e. linear regression) to see how the values in certain Xs are related to the value of Y, holding all else constant in the current sample dataset.
  - Predict values of Y for given values of Xs.
- Raw data -> data processing
  - Taking out outliers, checking for measurement errors, editing / altering data
- CSV = comma separated values

Data

- Previous method of data collection
  - Books (logbooks – Matthew Fontaine Maury)
  - Articles, newspapers..

- Big data
  - Behaviors are easily tracked / recorded
  - Once on the internet, always on the internet
  - 3 V
  - **volume, variety and velocity**

Quiz (True / False)

- Zip code is a quantitative variable.
- **False**

- Datetime is a quantitative variable.
- **False**

- Data is ready for analysis in its raw form.
- **False**

- We could change a qualitative variable into a quantitative variable in the data processing stage.
- **True**

B. Vectors and Data Frames

Data Structures

- Hadley Wikhan's book 'Advanced R' has a detailed treatment of data structures.

- R's base data structures can be organized by

- dimensionality
  - (1d, 2d, or nd)
- and whether they're **homogeneous** (all contents must be of the same type) or **heterogeneous** (the contents can be of different types).

Data Structures

- This gives rise to the five data types most often used in data analysis:

	Homogeneous	Heterogeneous
1d	Atomic Vector	List
2d	Matrix	Data frame
nd	Array	(Tensor Flow)

- Note that R has no 0-dimensional, or scalar types.

- Individual numbers (1,2,3..) or strings (Yes, No), which you might think would be scalars, are actually vectors of length one.

Data Structures

- For our class, we fill focus on 3 data types:
    - Atomic vectors
    - Lists
    - Data frames

Vectors

- Vector – a collection of data
- The basic data structure in R.
- Generally, a list of numbers or strings

Column Vector: $\begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}$

Row Vector: $\begin{pmatrix} 2 & 4 & 1 \end{pmatrix}$

- i.e.)
- A (vector): 1,2,3,4……10
- B (vector): Jane, Lisa, Mark, Alice, Lukas, ..
- C (vector): 3
- D (vector): No

Vectors

- *Atomic vectors*
- *Lists*

  - ***Common***
  - Type, typeof(), what it is
  - Length, length(), how many elements it contains
  - Attributes, attributes(), additional arbitrary metadata

  - ***Different***
  - The types of their elements:
    - All elements of an atomic vector must be **the same type**
    - Elements of a list can have **different types**

Atomic Vectors

- 4 common types of atomic vectors
  - Logical
  - Integer
  - Double (often called numeric)
  - Character
- 2 rare types
  - Complex
  - Raw
- Atomic vectors are usually created with c(), short for combine:

Atomic Vectors

- Example)
- *We can create a variable and assign it to the value using the assignment operator '<-'*
- dbl_var <- c(1, 2.5, 4.5)

- typeof(dbl_var)
- ## [1] "double"
- length(dbl_var)
- ## [1] 3
- attributes(dbl_var)
- ## NULL

Atomic Vectors

Price	Size	Lotsize	Bedrooms	Location	Sale date	Built date	Ocean view	Rooftop	Zipcode
400,000	10,000	20,000	NA	Long Beach, CA	2019/03/10	2012/01/28	no	yes	12345

- Missing values are specified with NA
- which is a logical vector of length 1
- NA will always be coerced to the correct type if used inside c()

- *dbl_var <- c(1, 2.5, NA, 4.5)*

Atomic Vectors

- Attributes
  - Defining ***factors***

- **Factor**
  - A vector that can contain only predefined values
  - Stores categorical data

M	F	M	M	F	M	F	F	M

- Factors are integer vectors using two attributes:
  - **class()** - which makes them behave differently from regular integer vectors
    - Class(X) of a factor vector X = “factor”
  - **levels()** - which defines the set of allowed values
    - Example) “m” and “f” for a ‘gender’ factor variable.

Atomic Vectors

- Factors are useful when you know the possible values a variable may take, even if you don't see all values in a given dataset
- Because factors are actually **integers** that look (and sometimes behave) like character vectors,
- Be careful when treating them like strings.
  - Some string methods coerce factors to strings, while others throw an error, and still others use the underlying integer values
  - For this reason, it's usually best to explicitly *convert factors to character vectors* if you need string-like behavior
- In early versions of R, there was a memory advantage to using factors instead of character vectors, but this is no longer the case

Atomic Vectors -> Matrix

- Atomic Vector – a collection of the same type of data
- **Matrix – a vector with 2 – dimensional shape information**

- Matrix : 2 dimensional shape information
  - Rows
  - Columns

- mat <- **matrix** (
- **c**(2 , 4, 3, 1, 5, 7), # the data elements
- nrow =2, # no. of rows
- ncol =3 # no. of columns
-)

```
> mat
     [,1] [,2] [,3]
[1,]    2    3    5
[2,]    4    1    7
```

Lists

- Different from atomic vectors because their elements can be of any type
- Use ***list()*** instead of ***c()***

- x <- list(1:3, "a", c(TRUE, FALSE, TRUE), c(2.3, 5.9))

- str(x)
- *str() is the function to check the structure of any R object*

- ## List of 4
- ## $: int [1:3] 1 2 3
- ## $: chr "a"
- ## $: logi [1:3] TRUE FALSE TRUE
- ## $: num [1:2] 2.3 5.9

Lists

- ***c()*** can combine several lists into one
  - If given a combination of atomic vectors and lists, c() coerces the vectors to lists before combining them

- Compare the results of list() and c():

Codes	
x <- list(list(1, 2), c(3, 4))	y <- c(list(1, 2), c(3, 4))
str(x)	str(y)
List of 2 $:List of 2 .. $: num 1 .. $: num 2 $: num [1:2] 3 4	List of 4 $: num 1 $: num 2 $: num 3 $: num 4

Lists

- Lists are used to build up many of the more complicated data structures in R, such as
- *Data frames*

Data Frames

- A data frame is the most common way of storing data in R

- Under the hood, a data frame is a list of equal-length vectors.
- This makes it a 2-dimensional structure, so it shares properties of both the matrix and the list.

- A data frame can be viewed as:
- a 1d structure (where it behaves like a list),
- or a 2d structure (where it behaves like a matrix).

Data Frames - Useful functions

- head(dataframe) # Top of dataframe
- tail(dataframe) # Bottom of dataframe
- dim(dataframe) # Dimensions
- ncol(dataframe) # Number of columns
- nrow(dataframe) # Number of rows
- names(dataframe) # Column names
- rownames(dataframe) # Row names
- str(dataframe) # Structure, with class, length, and content
- summary(dataframe) # Summary statistics for each columns

Data Frames - Useful functions

- Example)

- df <- data.frame(x = 1:3, y = c("a", "b", "c"))
- str(df)

- *## 'data.frame': 3 obs. of 2 variables:*
- *## $ x: int 1 2 3*
- *## $ y: Factor w/ 3 levels "a","b","c": 1 2 3*

- # Beware data.frame()'s default behaviour which turns strings into factors.
- # Use stringAsFactors = FALSE to suppress this behaviour:

- df <- data.frame(x = 1:3,y = c("a", "b", "c"),stringsAsFactors = FALSE)

What is a Data Frame in R?

- A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

Characteristics of a data frame:

- The column names should be non-empty.
- The row names should be unique.
- The data stored in a data frame can be of numeric, factor or character type.
- Each column should contain same number of data items.

Learning R : Creating a Data Frame

- # Create a data frame
- emp.data <- data.frame(emp_id = c (1:5), emp_name = c("Rick","Dan","Michelle","Ryan","Gary"), salary = c(623.3,515.2,611.0,729.0,843.25), start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11", "2015-03-27")), stringsAsFactors = FALSE)
- # Print this data frame
- print(emp.data)
- When we execute the above code, it produces the result:

```
  emp_id    emp_name     salary     start_date
1      1     Rick         623.30     2012-01-01
2      2     Dan          515.20     2013-09-23
3      3     Michelle     611.00     2014-11-15
4      4     Ryan         729.00     2014-05-11
5      5     Gary         843.25     2015-03-27
```

- See : Live Demo (Click Execute at top left. You can modify code)
- See other examples at tutorialspoint.com/r/r_data_frames.htm

End