wbs
WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Dr. Iman Ahmadi
Assistant Prof. of Marketing

Marketing & Strategy Analytics:

Exploratory Data Analysis I

Exercise 2.1 (A)

Replicate codes that we covered so far in RStudio!

- Material that we covered through wbsLive
- Material provided online

Exercise 2.1 (B)

Replicate the analysis in the following slides. The analysis aims to create sample data for a hypothetical retailer.

Describing Data: Univariate Analysis using ‘StoreData’ dataset

Picture from: https://www.which.co.uk/reviews/supermarkets/article/loyalty-cards-compared-a4ERY9a5NFJd (link)

Load the Data

- Set up a new script file (e.g., Session2.R) and clean up the current workspace: rm(list = ls())
- Load the StoreData file from My.WBS into R and name it as ‘store.df’

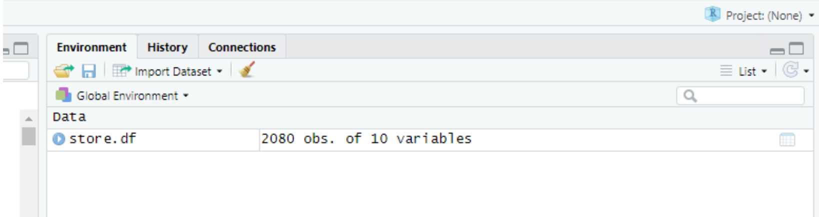
```
rm(list=ls())
store.df<-read.csv("D:/Warwick/.../StoreData.csv", header = TRUE)
```

- The dataset contains weekly (52 weeks per year) information (e.g., sales, price, promotion) about a few products collected over 20 stores over 2 years
- How many observations do you expect?

  2 x 52 x 20 = 2,080

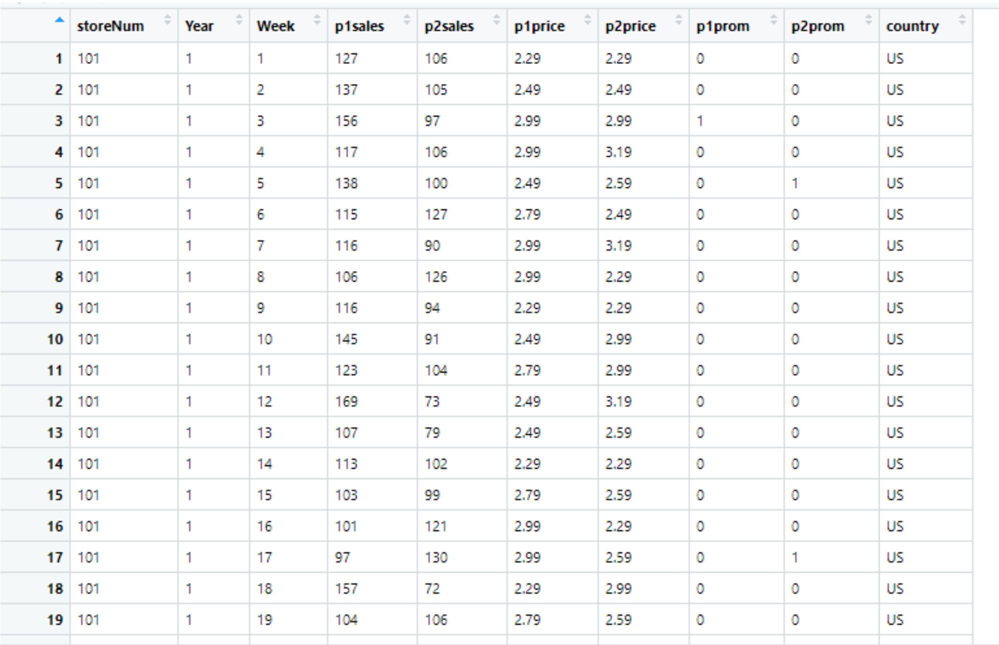
("StoreData" R code)

Screenshot of Dataset



Data

store.df 2080 obs. of 10 variables



	storeNum	Year	Week	p1sales	p2sales	p1price	p2price	p1prom	p2prom	country
1	101	1	1	127	106	2.29	2.29	0	0	US
2	101	1	2	137	105	2.49	2.49	0	0	US
3	101	1	3	156	97	2.99	2.99	1	0	US
4	101	1	4	117	106	2.99	3.19	0	0	US
5	101	1	5	138	100	2.49	2.59	0	1	US
6	101	1	6	115	127	2.79	2.49	0	0	US
7	101	1	7	116	90	2.99	3.19	0	0	US
8	101	1	8	106	126	2.99	2.29	0	0	US
9	101	1	9	116	94	2.29	2.29	0	0	US
10	101	1	10	145	91	2.49	2.99	0	0	US
11	101	1	11	123	104	2.79	2.99	0	0	US
12	101	1	12	169	73	2.49	3.19	0	0	US
13	101	1	13	107	79	2.49	2.59	0	0	US
14	101	1	14	113	102	2.29	2.29	0	0	US
15	101	1	15	103	99	2.79	2.59	0	0	US
16	101	1	16	101	121	2.99	2.29	0	0	US
17	101	1	17	97	130	2.99	2.59	0	1	US
18	101	1	18	157	72	2.29	2.99	0	0	US
19	101	1	19	104	106	2.79	2.59	0	0	US

Describing Data in R: Tables for One Variable

- Table() for counting

```
table(store.df$p1price)
```

```
2.19   2.29   2.49   2.79   2.99
395     444     423     443     375
```

What is the sum?

- The counts can be converted to proportions with prop.table()

```
prop.table(table(store.df$p1price))
```

```
     2.19        2.29        2.49        2.79        2.99
0.1899038  0.2134615  0.2033654  0.2129808  0.1802885
```

What is the sum?

Note: R can apply commands to commands!

Describing Data in R: Descriptive Functions (I/III)

- Core descriptive functions

Describe	Function	Value

Extremes	min(x)
	max(x)
		Minimum value
		Maximum value

Central Tendency	mean(x)
	median(x)
		Arithmetic mean
		Median

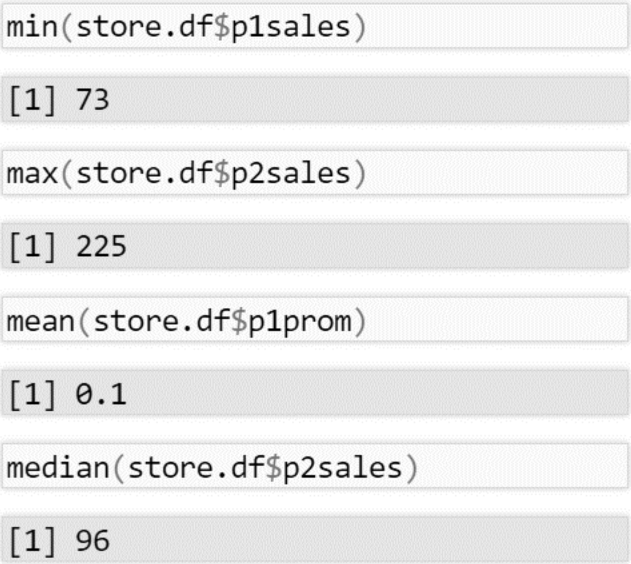
Dispersion	var(x)
	sd(x)
	IQR(x)
	mad(x)
		Variance around the mean
		Standard deviation
		Interquartile range, 25th-75th percentile
		Median absolute deviation (a robust variance estimator)

Points	quantile(x, probs=c(…))	

Note: ‘x’ is your variable!

Describing Data in R: Descriptive Functions (II/III)

- Core descriptive functions



```
min(store.df$p1sales)
```

```
[1] 73
```

```
max(store.df$p2sales)
```

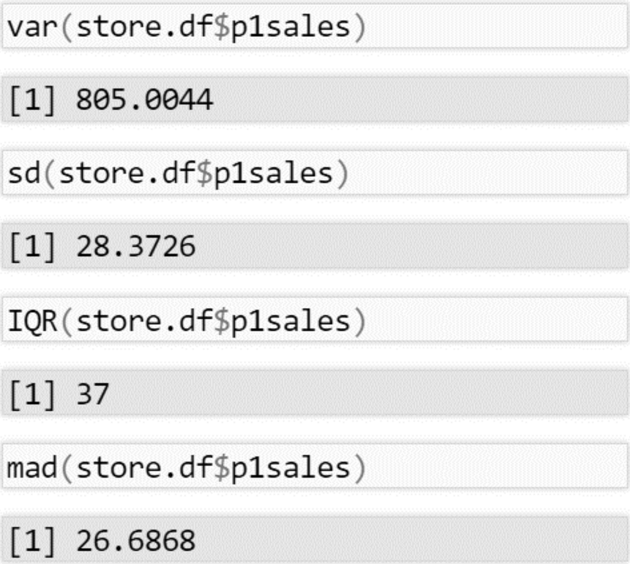
```
[1] 225
```

```
mean(store.df$p1prom)
```

```
[1] 0.1
```

```
median(store.df$p2sales)
```

```
[1] 96
```



```
var(store.df$p1sales)
```

```
[1] 805.0044
```

```
sd(store.df$p1sales)
```

```
[1] 28.3726
```

```
IQR(store.df$p1sales)
```

```
[1] 37
```

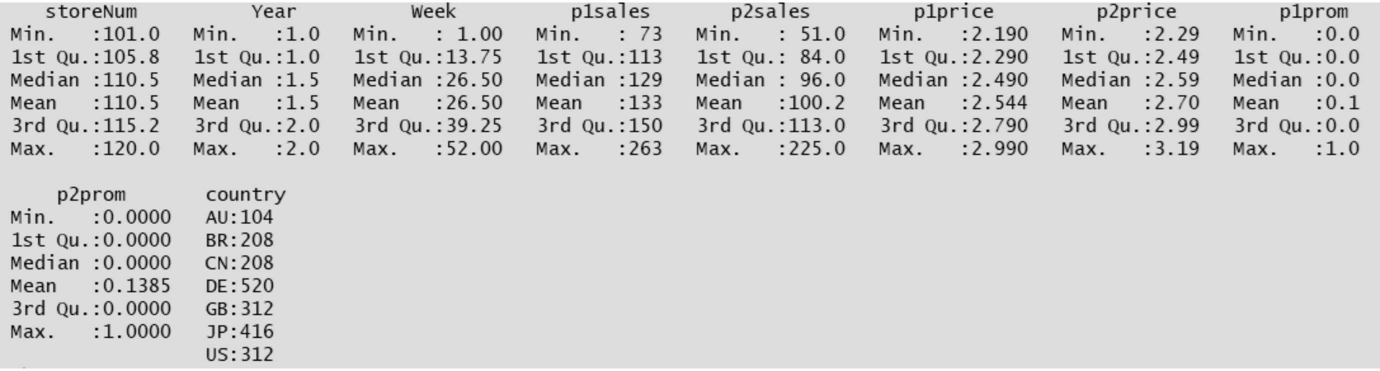
```
mad(store.df$p1sales)
```

```
[1] 26.6868
```

Describing Data in R: Descriptive Functions (III/III)

Note: many of these pieces of information can be obtained via summary()

```
summary(store.df)
```



```
    storeNum          Year          Week           p1sales        p2sales          p1price          p2price          p1prom   
 Min.   :101.0   Min.   :1.0   Min.   : 1.00   Min.   : 73   Min.   : 51.0   Min.   :2.190   Min.   :2.29   Min.   :0.0  
 1st Qu.:105.8   1st Qu.:1.0   1st Qu.:13.75   1st Qu.:113   1st Qu.: 84.0   1st Qu.:2.290   1st Qu.:2.49   1st Qu.:0.0  
 Median :110.5   Median :1.5   Median :26.50   Median :129   Median : 96.0   Median :2.490   Median :2.59   Median :0.0  
 Mean   :110.5   Mean   :1.5   Mean   :26.50   Mean   :133   Mean   :100.2   Mean   :2.544   Mean   :2.70   Mean   :0.1  
 3rd Qu.:115.2   3rd Qu.:2.0   3rd Qu.:39.25   3rd Qu.:150   3rd Qu.:113.0   3rd Qu.:2.790   3rd Qu.:2.99   3rd Qu.:0.0  
 Max.   :120.0   Max.   :2.0   Max.   :52.00   Max.   :263   Max.   :225.0   Max.   :2.990   Max.   :3.19   Max.   :1.0  

     p2prom         country
 Min.   :0.0000   AU:104   
 1st Qu.:0.0000   BR:208   
 Median :0.0000   CN:208   
 Mean   :0.1385   DE:520   
 3rd Qu.:0.0000   GB:312   
 Max.   :1.0000   JP:416   
                  US:312   
```

Describing Data in R: Two-Way Tables

- Note that tables index [row, column] like most things in R!

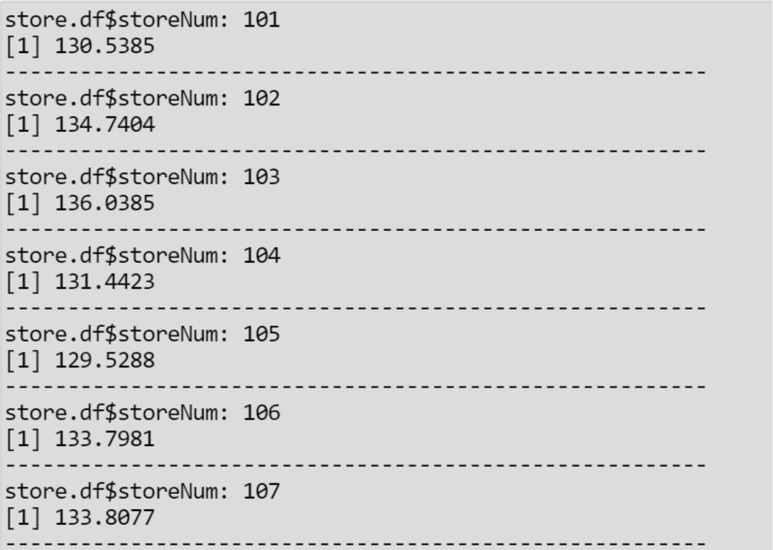
```
table(store.df$p1price, store.df$p1prom)
```

```
          0    1
2.19    354   41
2.29    398   46
2.49    381   42
2.79    396   47
2.99    343   32
```

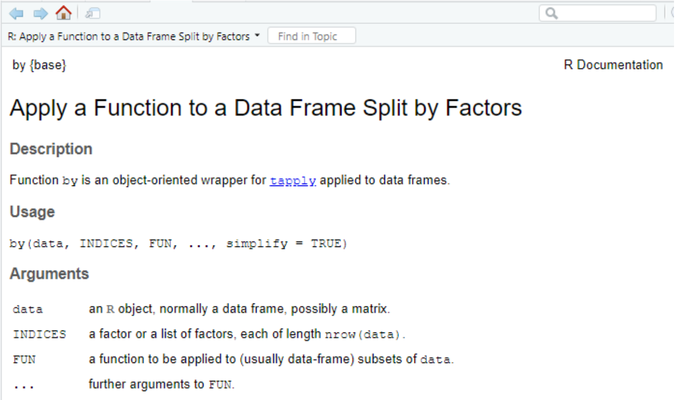
Describing Data in R: Descriptive Stats for Groups

- by() is one way to split data by a factor and apply a function to each group:

```
by(store.df$p1sales, store.df$storeNum, mean)
```



```
store.df$storeNum: 101
[1] 130.5385
--------------------------------------------------------
store.df$storeNum: 102
[1] 134.7404
--------------------------------------------------------
store.df$storeNum: 103
[1] 136.0385
--------------------------------------------------------
store.df$storeNum: 104
[1] 131.4423
--------------------------------------------------------
store.df$storeNum: 105
[1] 129.5288
--------------------------------------------------------
store.df$storeNum: 106
[1] 133.7981
--------------------------------------------------------
store.df$storeNum: 107
[1] 133.8077
--------------------------------------------------------
```



by {base} R Documentation

Apply a Function to a Data Frame Split by Factors

Description

Function `by` is an object-oriented wrapper for `tapply` applied to data frames.

Usage

```
by(data, INDICES, FUN, ..., simplify = TRUE)
```

Arguments

`data`	an R object, normally a data frame, possibly a matrix.
`INDICES`	a factor or a list of factors, each of length `nrow(data)`.
`FUN`	a function to be applied to (usually data-frame) subsets of `data`.
`...`	further arguments to `FUN`.

(try ‘?by’ in the Console)

Workshop Session:
Exercise 2.2 – Salaries for Professors

Access the Salaries data set:

```
library(car)    # install.packages("car") if needed; search how you can install a library in RStudio
data(Salaries)
```

1. How many variables and observations are there in the data set?
2. How many professors have more than 40 years of service?

(→ hint: you can sum() a logical vector)

3. How many have salary > $150000?
4. What is the mean salary for professors with >20 years service?
5. How do you find out more about the data set?

Note: by 'professors' we mean all three levels (i.e., "AsstProf","AssocProf", "Prof")

("Salary Prof" R code)

Fox J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition Sage

Thank You!

Iman.Ahmadi@wbs.ac.uk
Room No.: 3.207