기초 100문제

Print to PDF

Contents

Attention

아래 6가지 패키지 베이스로 문제 풀었습니다. 설치 후 진행해주세요 **dplyr, reshape2, stringr, tidyr, lubridate, zoo**

01 Getting & Knowing Data

Attention

롤 랭킹 데이터 : https://www.kaggle.com/datasnaek/league-of-legends

DataUrl = 'https://raw.githubusercontent.com/Datamanim/pandas/main/lol.csv'

Question 1

데이터를 로드하라. 데이터는 \t을 기준으로 구분되어있다.

```
df <- read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/lol.csv',sep='\t')
```

```
class(df)
```

'data.frame'

Question 2

데이터의 상위 5개 행을 출력하라

```
head(df)
```

	gameId	creationTime	gameDuration	seasonId	winner	firstBlood	firstTower	firstInhibitor	firstBaron	firstDragon	…	t2_towerKills	t2_inhibitorKills	t2_
	<dbl>	<dbl>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	…	<int>	<int>	
1	3326086514	1.504279e+12	1949	9	1	2	1	1	1	1	…	5	0	
2	3229566029	1.497849e+12	1851	9	1	1	1	1	0	1	…	2	0	
3	3327363504	1.504360e+12	1493	9	1	2	1	1	1	2	…	2	0	
4	3326856598	1.504349e+12	1758	9	1	1	1	1	1	1	…	0	0	
5	3330080762	1.504554e+12	2094	9	1	2	1	1	1	1	…	3	0	
6	3287435705	1.501668e+12	2059	9	1	2	2	1	1	2	…	6	0	

A data.frame: 6 × 61

Question 3

데이터의 행과 열의 갯수를 파악하라

```
dim(df)
```

51490 · 61

Question 4

전체 컬럼을 출력하라

```
colnames(df)
```

'gameId' · 'creationTime' · 'gameDuration' · 'seasonId' · 'winner' · 'firstBlood' · 'firstTower' · 'firstInhibitor' · 'firstBaron' · 'firstDragon' · 'firstRiftHerald' · 't1_champ1id' · 't1_champ1_sum1' · 't1_champ1_sum2' · 't1_champ2id' · 't1_champ2_sum1' · 't1_champ2_sum2' · 't1_champ3id' · 't1_champ3_sum1' · 't1_champ3_sum2' · 't1_champ4id' · 't1_champ4_sum1' · 't1_champ4_sum2' ·

't2_towerKills' · 't2_inhibitorKills' · 't2_baronKills' · 't2_dragonKills' · 't2_riftHeraldKills' · 't2_ban1' · 't2_ban2' · 't2_ban3' · 't2_ban4' · 't2_ban5'

Question 5

6번째 컬럼명을 출력하라

```
Ans <-names(df[c(6)])
print(Ans)
```

```
[1] "firstBlood"
```

Question 6

6번째 컬럼의 데이터 타입을 확인하라

```
Ans <- sapply(df[c(6)], class)
print(Ans)
```

```
firstBlood
 "integer"
```

Question 7

데이터셋의 인덱스 구성은 어떤가

```
Ans <- summary(rownames(df))
print(Ans)
```

```
   Length     Class      Mode
    51490 character character
```

Question 8

6번째 컬럼의 3번째 값은 무엇인가?

```
Ans <- df[3,6]
print(Ans)
```

```
[1] 2
```

Attention

제주 날씨,인구에 따른 교통량데이터 : 출처 제주 데이터 허브 **DataUrl =**
'https://raw.githubusercontent.com/Datamanim/pandas/main/Jeju.csv'

Question 9

데이터를 로드하라. 컬럼이 한글이기에 적절한 처리해줘야함

```
library(readr)
df =
read_csv("https://raw.githubusercontent.com/Datamanim/pandas/main/Jeju.csv",locale=locale(
encoding="EUC-KR"),show_col_types = FALSE)
class(df)
```

'spec_tbl_df' · 'tbl_df' · 'tbl' · 'data.frame'

Question 10

데이터 마지막 3개행을 출력하라

```
Ans<-tail(df,3)
Ans
```

id	일자	시도명	읍면동명	거주인구	근무인구	방문인구	총 유동인구	평균속도	평균 소요시간	평균기온	일강수량	평균풍속
<dbl>	<date>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
32066	2020-04-30	제주시	도두동	28397.48	3144.895	84052.7	115595.1	41.053	29.421	20.3	0	3.0
32067	2020-04-30	서귀포시	안덕면	348037.85	29106.286	251129.7	628273.8	46.595	49.189	17.6	0	3.5
32068	2020-04-30	제주시	연동	1010643.37	65673.477	447622.1	1523938.9	40.863	27.765	14.1	0	4.8

A tibble: 3 × 13

Question 11

```r
# install.packages("dplyr")  # Install dplyr
library(dplyr)           # Load dplyr

Ans <- select_if(df, is.numeric)
head(Ans)
```

```
Warning message:
“package ‘dplyr’ was built under R version 4.0.5”
```

```
Attaching package: ‘dplyr’
```

```
The following objects are masked from ‘package:stats’:

    filter, lag
```

```
The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

id	거주인구	근무인구	방문인구	총 유동인구	평균 속도	평균 소요 시간	평균 기온	일강수량	평균 풍속
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
22448	32249.99	3418.266	102709.09	138377.3	39.556	29.167	5.0	0	2.5
22449	213501.00	10341.172	112692.79	336535.0	32.900	30.900	5.0	0	2.5
22450	1212382.22	96920.834	541194.48	1850497.5	29.538	35.692	2.9	0	2.4
22451	33991.65	6034.253	72155.92	112181.8	30.000	23.500	2.9	0	2.4
22452	155036.92	9403.969	150882.41	315323.3	41.583	14.375	5.1	0	2.3
22453	119524.38	5616.131	77338.34	202478.9	38.222	28.000	4.9	0	1.9

A tibble: 6 × 10

Question 12

범주형 변수를 가진 컬럼만 필터하여 데이터프레임을 만들고 상위 5행을 출력하라

```r
# install.packages("dplyr")  # Install dplyr
library("dplyr")           # Load dplyr

Ans <- select_if(df, is.character)
head(Ans)
```

시도명	읍면동명
<chr>	<chr>
제주시	도두동
제주시	외도동
제주시	이도2동
제주시	일도1동
서귀포시	대천동
서귀포시	서홍동

A tibble: 6 × 2

Question 13

각 컬럼의 결측치 숫자를 파악하라

```r
Ans = colSums(is.na(df))
Ans
```

id: 0 **일자:** 0 **시도명:** 0 **읍면동명:** 0 **거주인구:** 0 **근무인구:** 0 **방문인구:** 0

총 유동인구: 0 **평균 속도:** 0 **평균 소요 시간:** 0 **평균 기온:** 0 **일강수량:** 0

평균 풍속: 0

Question 14

각 컬럼의 데이터수, 데이터타입을 한번에 확인하라

```r
# install.packages("psych")
library("psych")
Ans = describe(df)
Ans
```

```
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to min; returning Inf”
```

```
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to max; returning -Inf”
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	s
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl
id	1	9621	2.725800e+04	2.777488e+03	27258.00	2.725800e+04	3565.65300	22448.000	32068.000	9620.000	0.00000000	-1.2003742	2.831666e+0
일자	2	9621	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf	NA	NA	N.
시도명*	3	9621	1.599314e+00	4.900629e-01	2.00	1.624139e+00	0.00000	1.000	2.000	1.000	-0.40526907	-1.8359478	4.996222e-0
읍면동명*	4	9621	2.134809e+01	1.176629e+01	22.00	2.143381e+01	14.82600	1.000	41.000	40.000	-0.04626972	-1.1660758	1.199581e-0
거주인구	5	9621	3.174315e+05	2.982079e+05	222110.46	2.610080e+05	210521.47714	9305.552	1364503.913	1355198.361	1.54926880	1.7087217	3.040249e+0
근무인구	6	9621	3.547120e+04	4.038121e+04	21960.93	2.625729e+04	18357.90013	1407.936	263476.965	262069.029	2.52411926	6.5918962	4.116890e+0
방문인구	7	9621	1.958896e+05	1.407061e+05	152805.33	1.688951e+05	89397.99568	11538.322	723459.209	711920.887	1.60209993	1.9591824	1.434507e+0
총 유동인구	8	9621	5.487922e+05	4.608802e+05	386693.47	4.575345e+05	302944.59654	22251.810	2066483.867	2044232.057	1.62316887	1.9285258	4.698703e+0
평균속도	9	9621	4.110908e+01	8.758631e+00	39.64	4.095279e+01	11.12395	24.333	103.000	78.667	0.21585864	-0.7361091	8.929479e-0
평균소요시간	10	9621	3.721587e+01	1.299379e+01	34.50	3.655343e+01	14.08470	12.667	172.200	159.533	0.54943108	0.8556189	1.324725e-0
평균기온	11	9621	1.355083e+01	7.745515e+00	13.40	1.358641e+01	9.04386	-9.600	30.400	40.000	-0.03559627	-0.7960985	7.896601e-0
일강수량	12	9621	6.972426e+00	2.761726e+01	0.00	1.273733e+00	0.00000	0.000	587.500	587.500	9.15918621	120.9958377	2.815597e-0
평균풍속	13	9621	2.753171e+00	1.498538e+00	2.40	2.567136e+00	1.18608	0.000	13.333	13.333	1.67037647	4.8698740	1.527769e-0

A psych: 13 × 13

Question 15

각 수치형 변수의 분포(사분위, 평균, 최대 , 최소)를 확인하라

```
Ans = summary(df)
Ans
```

```
      id             일자                시도명            읍면동명
 Min.   :22448   Min.   :2018-01-01   Length:9621        Length:9621
 1st Qu.:24853   1st Qu.:2018-08-10   Class :character   Class :character
 Median :27258   Median :2019-03-23   Mode  :character   Mode  :character
 Mean   :27258   Mean   :2019-03-13
 3rd Qu.:29663   3rd Qu.:2019-10-13
 Max.   :32068   Max.   :2020-04-30
    거주인구          근무인구          방문인구       총 유동인구
 Min.   :   9306   Min.   :  1408   Min.   : 11538   Min.   :  22252
 1st Qu.:  95399   1st Qu.: 12074   1st Qu.: 99632   1st Qu.: 221691
 Median : 222110   Median : 21961   Median :152805   Median : 386694
 Mean   : 317432   Mean   : 35471   Mean   :195890   Mean   : 548792
 3rd Qu.: 410667   3rd Qu.: 40192   3rd Qu.:236325   3rd Qu.: 640692
 Max.   :1364504   Max.   :263477   Max.   :723459   Max.   :2066484
   평균 속도       평균 소요 시간     평균 기온        일강수량
 Min.   : 24.33   Min.   : 12.67   Min.   :-9.60   Min.   :  0.000
 1st Qu.: 34.25   1st Qu.: 27.89   1st Qu.: 7.60   1st Qu.:  0.000
 Median : 39.64   Median : 34.50   Median :13.40   Median :  0.000
 Mean   : 41.11   Mean   : 37.22   Mean   :13.55   Mean   :  6.972
 3rd Qu.: 49.10   3rd Qu.: 46.18   3rd Qu.:19.70   3rd Qu.:  1.500
 Max.   :103.00   Max.   :172.20   Max.   :30.40   Max.   :587.500
   평균 풍속
 Min.   : 0.000
 1st Qu.: 1.700
 Median : 2.400
 Mean   : 2.753
 3rd Qu.: 3.400
 Max.   :13.333
```

```
Ans = df['거주인구']
head(Ans)
```

거주인구
<dbl>
32249.99
213501.00
1212382.22
33991.65
155036.92
119524.38

A tibble: 6 × 1

Question 17

평균 속도 컬럼의 4분위 범위(IQR) 값을 구하여라

```
IQR(df$`평균 속도`)
```

14.855

Question 18

읍면동명 컬럼의 유일값 갯수를 출력하라

```
Ans <- length(unique(df$읍면동명))
print(Ans)
```

```
[1] 41
```

Question 19

읍면동명 컬럼의 유일값을 모두 출력하라

```
Ans <- unique(df$읍면동명)
print(Ans)
```

```
 [1] "도두동"  "외도동"  "이도2동" "일도1동" "대천동"  "서홍동"  "한경면"
 [8] "송산동"  "조천읍"  "일도2동" "영천동"  "예래동"  "대륜동"  "삼도1동"
[15] "이호동"  "건입동"  "중앙동"  "삼양동"  "삼도2동" "이도1동" "남원읍"
[22] "대정읍"  "정방동"  "효돈동"  "아라동"  "한림읍"  "구좌읍"  "용담1동"
[29] "오라동"  "화북동"  "연동"    "표선면"  "중문동"  "성산읍"  "안덕면"
[36] "천지동"  "노형동"  "동홍동"  "용담2동" "봉개동"  "애월읍"
```

02 Filtering & Sorting

Attention

식당데이터 : https://github.com/justmarkham/DAT8/blob/master/data/chipotle.tsv

DataUrl = 'https://raw.githubusercontent.com/Datamanim/pandas/main/chipo.csv'

Question 20

데이터를 로드하라.

```
df <- read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/chipo.csv',
na.strings=c(""))
Ans <- head(df)
Ans
```

	order_id	quantity	item_name	choice_description	item_price
	<int>	<int>	<chr>	<chr>	<chr>
1	1	1	Chips and Fresh Tomato Salsa	NA	$2.39
2	1	1	Izze	[Clementine]	$3.39
3	1	1	Nantucket Nectar	[Apple]	$3.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NA	$2.39
5	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98
6	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	$10.98

Question 21

quantity컬럼 값이 3인 데이터를 추출하여 첫 5행을 출력하라

```
Ans <- subset(df,quantity ==3)
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price
	<int>	<int>	<chr>	<chr>	<chr>
410	178	3	Chicken Bowl	[[Fresh Tomato Salsa (Mild), Tomatillo-Green Chili Salsa (Medium), Roasted Chili Corn Salsa (Medium)], [Black Beans, Rice, Fajita Veggies, Cheese, Guacamole, Lettuce]]	$32.94
446	193	3	Bowl	[Braised Carnitas, Pinto Beans, [Sour Cream, Cheese, Cilantro-Lime Rice]]	$22.20
690	284	3	Canned Soft Drink	[Diet Coke]	$3.75
819	338	3	Bottled Water	NA	$3.27
851	350	3	Canned Soft Drink	[Sprite]	$3.75
918	379	3	Canned Soft Drink	[Lemonade]	$3.75

A data.frame: 6 × 5

Question 22

quantity컬럼 값이 3인 데이터를 추출하여 index를 1부터 정렬하고 첫 5행을 출력하라

```
Ans <- subset(df,quantity ==3)
rownames(Ans) <- 1:nrow(Ans)
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price
	<int>	<int>	<chr>	<chr>	<chr>
1	178	3	Chicken Bowl	[[Fresh Tomato Salsa (Mild), Tomatillo-Green Chili Salsa (Medium), Roasted Chili Corn Salsa (Medium)], [Black Beans, Rice, Fajita Veggies, Cheese, Guacamole, Lettuce]]	$32.94
2	193	3	Bowl	[Braised Carnitas, Pinto Beans, [Sour Cream, Cheese, Cilantro-Lime Rice]]	$22.20
3	284	3	Canned Soft Drink	[Diet Coke]	$3.75
4	338	3	Bottled Water	NA	$3.27
5	350	3	Canned Soft Drink	[Sprite]	$3.75
6	379	3	Canned Soft Drink	[Lemonade]	$3.75

A data.frame: 6 × 5

Question 23

quantity , item_price 두개의 컬럼으로 구성된 새로운 데이터 프레임을 정의하라

```
new_df <- df[,c('quantity','item_price')]
head(new_df)
```

	quantity	item_price
	<int>	<chr>
1	1	$2.39
2	1	$3.39
3	1	$3.39
4	1	$2.39
5	2	$16.98
6	1	$10.98

A data.frame: 6 × 2

Question 24

원본데이터에서 item_price 컬럼의 달러표시 문자를 제거하고 float 타입으로 저장하여 new_price 컬럼에 저장하라

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1	1	Chips and Fresh Tomato Salsa	NA	$2.39	2.39
2	1	1	Izze	[Clementine]	$3.39	3.39
3	1	1	Nantucket Nectar	[Apple]	$3.39	3.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NA	$2.39	2.39
5	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98	16.98
6	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	$10.98	10.98

A data.frame: 6 × 6

Question 25

new_price 컬럼이 5이하의 값을 가지는 데이터프레임을 추출하고, 전체 갯수를 구하여라

```
t <-subset(df,new_price <=5)
Ans <- nrow(t)
print(Ans)
```

```
[1] 1652
```

Question 26

item_name명이 Chicken Salad Bowl 인 데이터 프레임을 추출하라고 index 값을 초기화 하여라

```
t<-subset(df,item_name =='Chicken Salad Bowl')
rownames(t)<- 1:nrow(t)
Ans <-t

head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	20	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto Beans, Lettuce]]	$8.75	8.75
2	60	2	Chicken Salad Bowl	[Tomatillo Green Chili Salsa, [Sour Cream, Cheese, Guacamole]]	$22.50	22.50
3	94	2	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto Beans, Guacamole]]	$22.50	22.50
4	111	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, Cheese, Sour Cream, Lettuce]]	$8.75	8.75
5	137	2	Chicken Salad Bowl	[Fresh Tomato Salsa, Fajita Vegetables]	$17.50	17.50
6	220	1	Chicken Salad Bowl	[Roasted Chili Corn Salsa, [Black Beans, Sour Cream, Cheese, Lettuce]]	$8.75	8.75

A data.frame: 6 × 6

Question 27

new_price값이 9 이하이고 item_name 값이 Chicken Salad Bowl 인 데이터 프레임을 추출하라

```
library(dplyr)

Ans <- df %>% filter(new_price <=9 & item_name =='Chicken Salad Bowl')
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	20	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto Beans, Lettuce]]	$8.75	8.75
2	111	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, Cheese, Sour Cream, Lettuce]]	$8.75	8.75
3	220	1	Chicken Salad Bowl	[Roasted Chili Corn Salsa, [Black Beans, Sour Cream, Cheese, Lettuce]]	$8.75	8.75
4	221	1	Chicken Salad Bowl	[Tomatillo Green Chili Salsa, [Fajita Vegetables, Black Beans, Pinto Beans, Lettuce]]	$8.75	8.75
5	221	1	Chicken Salad Bowl	[Tomatillo Green Chili Salsa, [Fajita Vegetables, Rice, Cheese, Sour Cream, Lettuce]]	$8.75	8.75
6	234	1	Chicken Salad Bowl	[Fresh Tomato Salsa, Fajita Vegetables]	$8.75	8.75

A data.frame: 6 × 6

Question 28

df의 new_price 컬럼 값에 따라 오름차순으로 정리하고 index를 초기화 하여라

```
Ans<-df[order(df$new_price),]
rownames(Ans) <- 1:nrow(Ans)

head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	14	1	Canned Soda	[Dr. Pepper]	$1.09	1.09
2	17	1	Bottled Water	NA	$1.09	1.09
3	24	1	Canned Soda	[Sprite]	$1.09	1.09
4	38	1	Bottled Water	NA	$1.09	1.09
5	47	1	Canned Soda	[Dr. Pepper]	$1.09	1.09
6	51	1	Canned Soda	[Diet Dr. Pepper]	$1.09	1.09

A data.frame: 6 × 6

Question 29

df의 item_name 컬럼 값중 Chips 포함하는 경우의 데이터를 출력하라

```
Ans<-df[grepl('Chips',df$item_name),]
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1	1	Chips and Fresh Tomato Salsa	NA	$2.39	2.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NA	$2.39	2.39
7	3	1	Side of Chips	NA	$1.69	1.69
11	5	1	Chips and Guacamole	NA	$4.45	4.45
15	7	1	Chips and Guacamole	NA	$4.45	4.45
16	8	1	Chips and Tomatillo-Green Chili Salsa	NA	$2.39	2.39

A data.frame: 6 × 6

Question 30

df의 짝수번째 컬럼만을 포함하는 데이터프레임을 출력하라

```
odd_col <-seq_len(ncol(df))%%2
Ans<-df[,odd_col == 0 ]
head(Ans)
```

	quantity	choice_description	new_price
	<int>	<chr>	<dbl>
1	1	NA	2.39
2	1	[Clementine]	3.39
3	1	[Apple]	3.39
4	1	NA	2.39
5	2	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	16.98
6	1	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	10.98

A data.frame: 6 × 3

Question 31

df의 new_price 컬럼 값에 따라 내림차순으로 정리하고 index를 초기화 하여라

```
Ans<- df[order(df$new_price,decreasing=TRUE),]
rownames(Ans) <- 1:nrow(Ans)
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1443	15	Chips and Fresh Tomato Salsa	NA	$44.25	44.25
2	1398	3	Carnitas Bowl	[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream, Guacamole, Lettuce]]	$35.25	35.25
3	511	4	Chicken Burrito	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Lettuce]]	$35.00	35.00
4	1443	4	Chicken Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Cheese, Sour Cream]]	$35.00	35.00
5	1443	3	Veggie Burrito	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream, Guacamole]]	$33.75	33.75
6	178	3	Chicken Bowl	[[Fresh Tomato Salsa (Mild), Tomatillo-Green Chili Salsa (Medium), Roasted Chili Corn Salsa (Medium)], [Black Beans, Rice, Fajita Veggies, Cheese, Guacamole, Lettuce]]	$32.94	32.94

A data.frame: 6 × 6

Question 32

df의 item_name 컬럼 값이 Steak Salad 또는 Bowl 인 데이터를 인덱싱하라

```
Ans<-subset(df,item_name =='Steak Salad' |item_name =='Bowl')
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
446	193	3	Bowl	[Braised Carnitas, Pinto Beans, [Sour Cream, Cheese, Cilantro-Lime Rice]]	$22.20	22.20
665	276	1	Steak Salad	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Fajita Veggies, Cheese, Lettuce]]	$8.99	8.99
674	279	1	Bowl	[Adobo-Marinated and Grilled Steak, [Sour Cream, Salsa, Cheese, Cilantro-Lime Rice, Guacamole]]	$7.40	7.40
753	311	1	Steak Salad	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Fajita Veggies, Cheese, Lettuce]]	$8.99	8.99
894	369	1	Steak Salad	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Lettuce]]	$8.99	8.99
3503	1406	1	Steak Salad	[[Lettuce, Fajita Veggies]]	$8.69	8.69

A data.frame: 6 × 6

Question 33

df의 item_name 컬럼 값이 Steak Salad 또는 Bowl 인 데이터를 데이터 프레임화 한 후, item_name를 기준으로 중복행이 있으면 제거하되 첫번째 케이스만 남겨라

```
sub<-subset(df,item_name =='Steak Salad' |item_name =='Bowl')
Ans<-distinct(sub,item_name,.keep_all =TRUE)
Ans
```

order_id	quantity	item_name	choice_description	item_price	new_price
<int>	<int>	<chr>	<chr>	<chr>	<dbl>
193	3	Bowl	[Braised Carnitas, Pinto Beans, [Sour Cream, Cheese, Cilantro-Lime Rice]]	$22.20	22.20
276	1	Steak Salad	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Fajita Veggies, Cheese, Lettuce]]	$8.99	8.99

A data.frame: 2 × 6

Question 34

df의 item_name 컬럼 값이 Steak Salad 또는 Bowl 인 데이터를 데이터 프레임화 한 후, item_name를 기준으로 중복행이 있으면 제거하되 마지막 케이스만 남겨라

```
sub<-subset(df,item_name =='Steak Salad' |item_name =='Bowl')
sub <- sub[order(-sub$order_id),]
Ans<- sub[!duplicated(sub$item_name),]
Ans
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
3503	1406	1	Steak Salad	[[Lettuce, Fajita Veggies]]	$8.69	8.69
674	279	1	Bowl	[Adobo-Marinated and Grilled Steak, [Sour Cream, Salsa, Cheese, Cilantro-Lime Rice, Guacamole]]	$7.40	7.40

A data.frame: 2 × 6

Question 35

df의 데이터 중 new_price값이 new_price값의 평균값 이상을 가지는 데이터들을 인덱싱하라 (dplyr 패키지의 filter 이용)

```
library(dplyr)

Ans<- filter(df,new_price >=mean(df$new_price))
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98	16.98
2	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	$10.98	10.98
3	4	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Fajita Vegetables, Black Beans, Pinto Beans, Cheese, Sour Cream, Guacamole, Lettuce]]	$11.75	11.75
4	4	1	Steak Soft Tacos	[Tomatillo Green Chili Salsa, [Pinto Beans, Cheese, Sour Cream, Lettuce]]	$9.25	9.25
5	5	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Pinto Beans, Cheese, Sour Cream, Lettuce]]	$9.25	9.25
6	6	1	Chicken Crispy Tacos	[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream]]	$8.75	8.75

A data.frame: 6 × 6

Question 36

df의 데이터 중 item_name의 값이 Izze 데이터를 Fizzy Lizzy로 수정하라

```
# install.packages("stringr")
library(stringr)

df$item_name<- str_replace(df$item_name,'Izze','Fizzy Lizzy')
head(df)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1	1	Chips and Fresh Tomato Salsa	NA	$2.39	2.39
2	1	1	Fizzy Lizzy	[Clementine]	$3.39	3.39
3	1	1	Nantucket Nectar	[Apple]	$3.39	3.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NA	$2.39	2.39
5	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98	16.98
6	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	$10.98	10.98

A data.frame: 6 × 6

Question 37

df의 데이터 중 choice_description 값이 NaN 인 데이터의 갯수를 구하여라

```
Ans<-sum(is.na(df$choice_description))
Ans
```

1246

Question 38

df의 데이터 중 choice_description 값이 NaN 인 데이터를 NoData 값으로 대체하라

```
df$choice_description[is.na(df$choice_description)] = 'NoData'
head(df)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1	1	Chips and Fresh Tomato Salsa	NoData	$2.39	2.39
2	1	1	Fizzy Lizzy	[Clementine]	$3.39	3.39
3	1	1	Nantucket Nectar	[Apple]	$3.39	3.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NoData	$2.39	2.39
5	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98	16.98
6	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	$10.98	10.98

A data.frame: 6 × 6

Question 39

df의 데이터 중 choice_description 값에 Black이 들어가는 경우를 인덱싱하라

```
Ans<-df[grepl('Black',df$choice_description),]
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
5	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98	16.98
8	4	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Fajita Vegetables, Black Beans, Pinto Beans, Cheese, Sour Cream, Guacamole, Lettuce]]	$11.75	11.75
10	5	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Pinto Beans, Cheese, Sour Cream, Lettuce]]	$9.25	9.25
12	6	1	Chicken Crispy Tacos	[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream]]	$8.75	8.75
13	6	1	Chicken Soft Tacos	[Roasted Chili Corn Salsa, [Rice, Black Beans, Cheese, Sour Cream]]	$8.75	8.75
18	9	1	Chicken Burrito	[Fresh Tomato Salsa (Mild), [Black Beans, Rice, Cheese, Sour Cream, Lettuce]]	$8.49	8.49

A data.frame: 6 × 6

Question 40

df의 데이터 중 choice_description 값에 Vegetables 들어가지 않는 경우의 갯수를 출력하라

```
sub<-df[!grepl('Vegetables',df$choice_description),]
Ans<-nrow(sub)
print(Ans)
```

```
[1] 3900
```

Question 41

df의 데이터 중 item_name 값이 N으로 시작하는 데이터를 모두 추출하라

```
Ans<- df[grepl('^N',df$item_name),] ## 정규표현식
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
3	1	1	Nantucket Nectar	[Apple]	$3.39	3.39
23	11	1	Nantucket Nectar	[Pomegranate Cherry]	$3.39	3.39
106	46	1	Nantucket Nectar	[Pineapple Orange Banana]	$3.39	3.39
174	77	1	Nantucket Nectar	[Apple]	$3.39	3.39
206	91	1	Nantucket Nectar	[Peach Orange]	$3.39	3.39
437	189	1	Nantucket Nectar	[Pomegranate Cherry]	$3.39	3.39

A data.frame: 6 × 6

Question 42

df의 데이터 중 item_name 값의 단어갯수가 15개 이상인 데이터를 인덱싱하라

```
Ans<- subset(df,nchar(df$item_name)>=15)
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1	1	Chips and Fresh Tomato Salsa	NoData	$2.39	2.39
3	1	1	Nantucket Nectar	[Apple]	$3.39	3.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NoData	$2.39	2.39
9	4	1	Steak Soft Tacos	[Tomatillo Green Chili Salsa, [Pinto Beans, Cheese, Sour Cream, Lettuce]]	$9.25	9.25
11	5	1	Chips and Guacamole	NoData	$4.45	4.45
12	6	1	Chicken Crispy Tacos	[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream]]	$8.75	8.75

A data.frame: 6 × 6

Question 43

df의 데이터 중 new_price값이 아래 lst에 해당하는 경우의 데이터 프레임을 구하고 그 갯수를 출력하라 lst =[1.69, 2.39, 3.39, 4.45, 9.25, 10.98, 11.75, 16.98]

```
lst <- c(1.69, 2.39, 3.39, 4.45, 9.25, 10.98, 11.75, 16.98)
Ans<-subset(df,new_price %in% lst)
head(Ans)
```

	order_id	quantity	item_name	choice_description	item_price	new_price
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	1	1	Chips and Fresh Tomato Salsa	NoData	$2.39	2.39
2	1	1	Fizzy Lizzy	[Clementine]	$3.39	3.39
3	1	1	Nantucket Nectar	[Apple]	$3.39	3.39
4	1	1	Chips and Tomatillo-Green Chili Salsa	NoData	$2.39	2.39
5	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	$16.98	16.98
6	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sour Cream, Guacamole, Lettuce]]	$10.98	10.98

A data.frame: 6 × 6

03_Grouping

Attention

뉴욕 airBnB : https://www.kaggle.com/ptoscano230382/air-bnb-ny-2019 **DataUrl =** **'https://raw.githubusercontent.com/Datamanim/pandas/main/AB_NYC_2019.csv'**

Question 44

데이터를 로드하고 상위 5개 컬럼을 출력하라

```
df<- read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/AB_NYC_2019.csv')
head(df)
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_review
	<int>	<chr>	<int>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<chr>	<int>	<int>	<int>
1	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	
2	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	4
3	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
4	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	27
5	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	
6	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray Hill	40.74767	-73.97500	Entire home/apt	200	3	7

A data.frame: 6 × 16

Question 45

데이터의 각 host_name의 빈도수를 구하고 host_name으로 정렬하여 상위 5개를 출력하라

```
library(dplyr)

Ans <- head(count(df,host_name,sort=TRUE))
Ans
```

	host_name	n
	<chr>	<int>
1	Michael	417
2	David	403
3	Sonder (NYC)	327
4	John	294
5	Alex	279
6	Blueground	232

A data.frame: 6 × 2

Question 46

데이터의 각 host_name의 빈도수를 구하고 빈도수 기준 내림차순 정렬한 데이터 프레임을 만들어라. 빈도수 컬럼은 counts로 명명하라

```
library(dplyr)
Ans <- head(count(df,host_name,sort=TRUE))
colnames(Ans) <- c('host_name','counts')
head(Ans)
```

	host_name	counts
	<chr>	<int>
1	Michael	417
2	David	403
3	Sonder (NYC)	327
4	John	294
5	Alex	279
6	Blueground	232

A data.frame: 6 × 2

Question 47

neighbourhood_group의 값에 따른 neighbourhood컬럼 값의 unique한 갯수를 구하여라

```
library(dplyr)

Ans<- df %>%
    group_by(neighbourhood_group,neighbourhood) %>%
    count()

head(Ans)
```

neighbourhood_group	neighbourhood	n
<chr>	<chr>	<int>
Bronx	Allerton	42
Bronx	Baychester	7
Bronx	Belmont	24
Bronx	Bronxdale	19
Bronx	Castle Hill	9
Bronx	City Island	18

A grouped_df: 6 × 3

Question 48

neighbourhood_group의 값에 따른 neighbourhood컬럼 값들의 unique한 갯수 중 neighbourhood_group그룹을 기준으로 최댓값을 가지는 데이터들을 출력하라

```
Ans <-df %>%
    group_by(neighbourhood_group,neighbourhood) %>%
    count() %>%
    group_by(neighbourhood_group) %>%
    slice(which.max(n))

Ans
```

neighbourhood_group	neighbourhood	n
<chr>	<chr>	<int>
Bronx	Kingsbridge	70
Brooklyn	Williamsburg	3920
Manhattan	Harlem	2658
Queens	Astoria	900
Staten Island	St. George	48

A grouped_df: 5 × 3

Question 49

neighbourhood_group 값에 따른 price값의 평균, 분산, 최대, 최소 값을 구하여라

```
Ans<-df %>%
    group_by(neighbourhood_group) %>%
    summarize(mean = mean(price)
              ,var = var(price)
              ,std=sd(price)
              ,max=max(price)
              ,min=min(price))

Ans
```

neighbourhood_group	mean	var	std	max	min
<chr>	<dbl>	<dbl>	<dbl>	<int>	<int>
Bronx	87.49679	11386.89	106.7093	2500	0
Brooklyn	124.38321	34921.72	186.8735	10000	0
Manhattan	196.87581	84904.16	291.3832	10000	0
Queens	99.51765	27923.13	167.1022	10000	10
Staten Island	114.81233	77073.09	277.6204	5000	13

A tibble: 5 × 6

Question 50

neighbourhood_group 값에 따른 reviews_per_month 평균, 분산, 최대, 최소 값을 구하여라

```
Ans<-df %>%
    group_by(neighbourhood_group) %>%
    summarize(mean = mean(reviews_per_month, na.rm = TRUE)
              ,var = var(reviews_per_month, na.rm = TRUE)
              ,std=sd(reviews_per_month, na.rm = TRUE)
              ,max=max(reviews_per_month, na.rm = TRUE)
              ,min=min(reviews_per_month, na.rm = TRUE))

Ans
```

neighbourhood_group	mean	var	std	max	min
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Bronx	1.837831	2.799878	1.673284	10.34	0.02
Brooklyn	1.283212	2.299040	1.516259	14.00	0.01
Manhattan	1.272131	2.651206	1.628252	58.50	0.01

Question 51

neighbourhood 값과 neighbourhood_group 값에 따른 price 의 평균을 구하라

```
Ans<- df %>%
    group_by(neighbourhood,neighbourhood_group) %>%
    summarize(mean = mean(price),.groups = "drop_last")
head(Ans)
```

neighbourhood	neighbourhood_group	mean
<chr>	<chr>	<dbl>
Allerton	Bronx	87.59524
Arden Heights	Staten Island	67.25000
Arrochar	Staten Island	115.00000
Arverne	Queens	171.77922
Astoria	Queens	117.18778
Bath Beach	Brooklyn	81.76471

A grouped_df: 6 × 3

Question 52

neighbourhood 값과 neighbourhood_group 값에 따른 price 의 평균을 계층적 indexing 없이 구하라

```
# install.packages('reshape')
library(reshape)

Ans<- df %>%
    group_by(neighbourhood,neighbourhood_group) %>%
    summarize(mean = mean(price),.groups = "drop_last")

res <-cast(data = Ans, neighbourhood ~ neighbourhood_group,value='mean')
head(res)
```

```
Attaching package: ‘reshape’
```

```
The following object is masked from ‘package:dplyr’:

    rename
```

	neighbourhood	Bronx	Brooklyn	Manhattan	Queens	Staten Island
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Allerton	87.59524	NA	NA	NA	NA
2	Arden Heights	NA	NA	NA	NA	67.25
3	Arrochar	NA	NA	NA	NA	115.00
4	Arverne	NA	NA	NA	171.7792	NA
5	Astoria	NA	NA	NA	117.1878	NA
6	Bath Beach	NA	81.76471	NA	NA	NA

A cast_df: 6 × 6

Question 53

neighbourhood 값과 neighbourhood_group 값에 따른 price 의 평균을 계층적 indexing 없이 구하고 nan 값은 -999값으로 채워라

```
# install.packages('reshape')
library(reshape)

Ans<- df %>%
    group_by(neighbourhood,neighbourhood_group) %>%
    summarize(mean = mean(price),.groups = "drop_last")

res <-cast(data = Ans, neighbourhood ~ neighbourhood_group,fill=-999,value='mean')
head(res)
```

	neighbourhood	Bronx	Brooklyn	Manhattan	Queens	Staten Island
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Allerton	87.59524	-999.00000	-999	-999.0000	-999.00
2	Arden Heights	-999.00000	-999.00000	-999	-999.0000	67.25
3	Arrochar	-999.00000	-999.00000	-999	-999.0000	115.00
4	Arverne	-999.00000	-999.00000	-999	171.7792	-999.00
5	Astoria	-999.00000	-999.00000	-999	117.1878	-999.00
6	Bath Beach	-999.00000	81.76471	-999	-999.0000	-999.00

A cast_df: 6 × 6

Question 54

데이터중 neighbourhood_group 값이 Queens값을 가지는 데이터들 중 neighbourhood 그룹별로 price값의 평균, 분산, 최댓값, 최소값을

```
library(dplyr)
Ans<-df %>%
    filter(neighbourhood_group =='Queens') %>%
    group_by(neighbourhood) %>%
    summarize(mean = mean(price)
            ,var = var(price, na.rm = TRUE)
            ,std=sd(price, na.rm = TRUE)
            ,max=max(price, na.rm = TRUE)
            ,min=min(price, na.rm = TRUE))

head(Ans)
```

neighbourhood	mean	var	std	max	min
<chr>	<dbl>	<dbl>	<dbl>	<int>	<int>
Arverne	171.77922	37383.411	193.34790	1500	35
Astoria	117.18778	122428.811	349.89829	10000	25
Bay Terrace	142.00000	6816.400	82.56149	258	32
Bayside	157.94872	166106.471	407.56162	2600	30
Bayswater	87.47059	2330.890	48.27929	230	45
Belle Harbor	171.50000	8226.571	90.70045	350	85

A tibble: 6 × 6

Question 55

데이터중 neighbourhood_group 값에 따른 room_type 컬럼의 unique value의 각 갯수를 구하고 neighbourhood_group 값을 기준으로 각 값의 비율을 구하여라

```
library(dplyr)
ra <-df %>%
    group_by(neighbourhood_group,room_type) %>%
    count()


res <-cast(data = ra, neighbourhood_group ~ room_type,value='n')
Ans <-cbind(res[,1],prop.table(data.matrix(res[,c(2:4)]),1))

Ans
```

		Entire home/apt	Private room	Shared room
1	Bronx	0.347387717690192	0.597616865261228	0.0549954170485793
2	Brooklyn	0.475477516912057	0.503979307600477	0.0205431754874652
3	Manhattan	0.609343982272287	0.368496375975255	0.0221596417524583
4	Queens	0.369925873632192	0.595128838686904	0.0349452876809036
5	Staten Island	0.471849865951743	0.50402144772118	0.0241286863270777

A matrix: 5 × 4 of type chr

04_Apply , Map

Attention

카드이용데이터 : https://www.kaggle.com/sakshigoyal7/credit-card-customers **DataUrl = 'https://raw.githubusercontent.com/Datamanim/pandas/main/BankChurnersUp.csv'**

Question 56

데이터를 로드하고 데이터 행과 열의 갯수를 출력하라

```
df <-
read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/BankChurnersUp.csv')
dim(df)
```

10127 · 19

Question 57

Income_Category의 카테고리를 stringr패키지의 str_replace 함수를 이용하여 다음과 같이 변경하여 newIncome 컬럼에 매핑하라

Unknown : N

Less than $40K : a

$40K - $60K : b

$60K - $80K : c

$80K - $120K : d

$120K +' : e

```
library(stringr)

df$newIncome = str_replace(df$Income_Category,'Less than \\$40K','a')
df$newIncome = str_replace(df$newIncome,'\\$40K - \\$60K','b')
df$newIncome = str_replace(df$newIncome,'\\$60K - \\$80K','c')
df$newIncome = str_replace(df$newIncome,'\\$80K - \\$120K','d')
df$newIncome = str_replace(df$newIncome,'\\$120K \\+','e')
df$newIncome = str_replace(df$newIncome,'Unknown','N')
```

	Income_Category	newIncome
	<chr>	<chr>
1	$60K - $80K	c
2	Less than $40K	a
3	$80K - $120K	d
4	Less than $40K	a
5	$60K - $80K	c
6	$40K - $60K	b

A data.frame: 6 × 2

Question 59

Customer_Age의 값을 이용하여 나이 구간을 AgeState 컬럼으로 정의하라. (0~9 : 0 , 10~19 :10 , 20~29 :20 ... 각 구간의 빈도수를 출력하라

```
library(dplyr)
df$AgeState <- df$Customer_Age %%10 *10
Ans<-count(df, AgeState)
Ans
```

AgeState	n
<dbl>	<int>
0	1011
10	961
20	1001
30	1053
40	996
50	1050
60	1053
70	998
80	963
90	1041

A data.frame: 10 × 2

Question 60

Education_Level의 값중 Graduate단어가 포함되는 값은 1 그렇지 않은 경우에는 0으로 변경하여 newEduLevel 컬럼을 정의하고 빈도수를 출력하라

```
library(stringr)
library(dplyr)

df$newEduLevel <- lapply(str_detect("Graduate",df$Education_Level),as.numeric)
count(df,newEduLevel)
```

newEduLevel	n
<list>	<int>
0	6999
1	3128

A data.frame: 2 × 2

Question 61

Credit_Limit 컬럼값이 4500 이상인 경우 1 그외의 경우에는 모두 0으로 하는 newLimit 정의하라. newLimit 각 값들의 빈도수를 출력하라

```
df$newLimit <- lapply(df$Credit_Limit >=4500,as.numeric)
count(df,newLimit)
```

newLimit	n
<list>	<int>
1	5096
0	5031

A data.frame: 2 × 2

Question 62

Marital_Status 컬럼값이 Married 이고 Card_Category 컬럼의 값이 Platinum인 경우 1 그외의 경우에는 모두 0으로 하는 newState컬럼

```
df$newState <- lapply(df$Marital_Status =='Married'& df$Card_Category
=='Platinum',as.numeric)
count(df,newState)
```

newState	n
<list>	<int>
0	10120
1	7

A data.frame: 2 × 2

Question 63

Gender 컬럼값 M인 경우 male F인 경우 female로 값을 변경하여 Gender 컬럼에 새롭게 정의하라. 각 value의 빈도를 출력하라

```
library(stringr)
library(dplyr)
df$Gender = str_replace(df$Gender,'M','male')
df$Gender = str_replace(df$Gender,'F','female')
count(df,Gender)
```

Gender	n
<chr>	<int>
female	5358
male	4769

A data.frame: 2 × 2

05_Time_Series

Attention

주가 데이터 : https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/06_Stats/Wind_Stats/wind.data **DataUrl = 'https://raw.githubusercontent.com/Datamanim/pandas/main/timeTest.csv'**

Question 64

데이터를 로드하고 첫 5행을 출력하라

```
df<-read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/timeTest.csv')
head(df)
```

	Yr_Mo_Dy	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2061-01-01	15.04	14.96	13.17	9.29	NA	9.87	13.67	10.25	10.83	12.58	18.50	15.04
2	2061-01-02	14.71	NA	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83
3	2061-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NA	8.50	7.67	12.75	12.71
4	2061-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88
5	2061-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83
6	2061-01-06	13.21	8.12	9.96	6.67	5.37	4.50	10.67	4.42	7.17	7.50	8.12	13.17

A data.frame: 6 × 13

Question 65

Yr_Mo_Dy을 date타입으로 변경하라

```
df$Yr_Mo_Dy<-as.Date(df$Yr_Mo_Dy)
head(df)
```

	Yr_Mo_Dy	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2061-01-01	15.04	14.96	13.17	9.29	NA	9.87	13.67	10.25	10.83	12.58	18.50	15.04
2	2061-01-02	14.71	NA	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83
3	2061-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NA	8.50	7.67	12.75	12.71
4	2061-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88
5	2061-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83
6	2061-01-06	13.21	8.12	9.96	6.67	5.37	4.50	10.67	4.42	7.17	7.50	8.12	13.17

A data.frame: 6 × 13

Question 66

Yr_Mo_Dy에 존재하는 년도의 유일값을 모두 출력하라

```
Ans<-unique(format(df$Yr_Mo_Dy,'%Y'))
Ans
```

'2061' · '2062' · '2063' · '2064' · '2065' · '2066' · '2067' · '2068' · '2069' · '2070' · '1971' · '1972' · '1973' · '1974' · '1975' · '1976' · '1977' · '1978'

Question 67

Yr_Mo_Dy에 년도가 2061년 이상의 경우에는 모두 잘못된 데이터이다. 해당경우의 값은 년도에서 100을 빼서 새롭게 날짜를 Yr_Mo_Dy 컬럼에 정의하라

```
# install.packages('lubridate')
library(lubridate)
library(dplyr)
```

```
Attaching package: ‘lubridate’
```

```
The following object is masked from ‘package:reshape’:

    stamp
```

```
The following objects are masked from ‘package:base’:

    date, intersect, setdiff, union
```

```
df$Yr_Mo_Dy<- if_else(format(df$Yr_Mo_Dy,'%Y') >=2061,df$Yr_Mo_Dy-years(100),df$Yr_Mo_Dy)
head(df)
```

	Yr_Mo_Dy	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1961-01-01	15.04	14.96	13.17	9.29	NA	9.87	13.67	10.25	10.83	12.58	18.50	15.04
2	1961-01-02	14.71	NA	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83
3	1961-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NA	8.50	7.67	12.75	12.71
4	1961-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88
5	1961-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83
6	1961-01-06	13.21	8.12	9.96	6.67	5.37	4.50	10.67	4.42	7.17	7.50	8.12	13.17

A data.frame: 6 × 13

Question 68

년도별 각컬럼의 평균값을 구하여라

```
Ans<-df %>%
    group_by(year=format(df$Yr_Mo_Dy,'%Y')) %>%
    summarise_all("mean", na.rm = TRUE) %>%
    select(-Yr_Mo_Dy)

head(Ans)
```

year	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1961	12.29958	10.35180	11.36237	6.958227	10.88176	7.729726	9.733923	8.858788	8.647652	9.835577	13.50279	13.68077
1962	12.24692	10.11044	11.73271	6.960440	10.65792	7.393068	11.020712	8.793753	8.316822	9.676247	12.93068	14.32396
1963	12.81345	10.83699	12.54115	7.330055	11.72411	8.434712	11.075699	10.336548	8.903589	10.224438	13.63888	14.99901
1964	12.36366	10.92016	12.10437	6.787787	11.45448	7.570874	10.259153	9.467350	7.789016	10.207951	13.74055	14.91030
1965	12.45137	11.07553	11.84877	6.858466	11.02479	7.478110	10.618712	8.879918	7.907425	9.918082	12.96425	15.59164
1966	13.46197	11.55721	12.02063	7.345726	11.80504	7.793671	10.579808	8.835096	8.514438	9.768959	14.26584	16.30726

A tibble: 6 × 13

Question 69

weekday컬럼을 만들고 요일별로 매핑하라 (일요일: 1 ~ 토요일 :7)

```
df$weekday <- wday(df$Yr_Mo_Dy)
head(df)
```

	Yr_Mo_Dy	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL	weekday
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1961-01-01	15.04	14.96	13.17	9.29	NA	9.87	13.67	10.25	10.83	12.58	18.50	15.04	1
2	1961-01-02	14.71	NA	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83	2
3	1961-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NA	8.50	7.67	12.75	12.71	3
4	1961-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88	4
5	1961-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83	5
6	1961-01-06	13.21	8.12	9.96	6.67	5.37	4.50	10.67	4.42	7.17	7.50	8.12	13.17	6

A data.frame: 6 × 14

Question 70

weekday컬럼을 기준으로 주말이면 1 평일이면 0의 값을 가지는 WeekCheck 컬럼을 만들어라

```
w <- c(1,7)
df$WeekCheck <- lapply(df$weekday %in% w,as.numeric)
head(df)
```

	Yr_Mo_Dy	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL	weekday	WeekCheck
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<list>
1	1961-01-01	15.04	14.96	13.17	9.29	NA	9.87	13.67	10.25	10.83	12.58	18.50	15.04	1	1
2	1961-01-02	14.71	NA	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83	2	0
3	1961-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NA	8.50	7.67	12.75	12.71	3	0
4	1961-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88	4	0
5	1961-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83	5	0
6	1961-01-06	13.21	8.12	9.96	6.67	5.37	4.50	10.67	4.42	7.17	7.50	8.12	13.17	6	0

A data.frame: 6 × 15

Question 71

년도, 일자 상관없이 모든 컬럼의 각 달의 평균을 구하여라

```
Ans<-df %>%
    select(-WeekCheck) %>%
    group_by(year=format(df$Yr_Mo_Dy,'%m')) %>%
    summarise_all("mean", na.rm = TRUE) %>%
    select(-Yr_Mo_Dy)

head(Ans)
```

year	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL	weekday
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
01	14.84732	12.914560	13.29962	7.199498	11.667734	8.054839	11.819355	9.512047	9.543208	10.053566	14.55052	18.02876	3.985663
02	13.71091	12.111122	12.87913	6.942411	11.551772	7.633858	11.206024	9.341437	9.313169	9.518051	13.72890	17.15614	4.000000
03	13.15869	11.505842	12.64812	7.265907	11.554516	7.959409	11.310179	9.635896	9.700324	10.096953	13.81061	16.90932	4.008961
04	12.55565	10.429759	12.20481	6.898037	10.677667	7.441389	10.221315	8.909056	8.930870	9.158019	12.66476	14.93761	4.000000
05	11.72403	10.145619	11.55039	6.307487	10.224301	6.942061	8.797738	8.452903	8.040806	8.524857	12.76726	13.73604	3.991039
06	10.45132	8.949704	10.36131	5.652278	9.529926	6.410093	8.009556	7.920796	7.639796	7.729185	12.24641	12.86182	4.009259

A tibble: 6 × 14

Question 72

모든 결측치는 컬럼기준 직전의 값으로 대체하고 첫번째 행에 결측치가 있을경우 뒤에있는 값으로 대채하라

```
library(tidyr)

Ans<-df %>%
    fill(everything(),.direction='down') %>%
    fill(everything(),.direction='up')
head(Ans)
```

```
Attaching package: ‘tidyr’
```

```
The following objects are masked from ‘package:reshape’:

    expand, smiths
```

	Yr_Mo_Dy	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL	weekday	WeekCheck
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<list>
1	1961-01-01	15.04	14.96	13.17	9.29	12.62	9.87	13.67	10.25	10.83	12.58	18.50	15.04	1	1
2	1961-01-02	14.71	14.96	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83	2	0
3	1961-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	10.04	8.50	7.67	12.75	12.71	3	0
4	1961-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88	4	0
5	1961-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83	5	0
6	1961-01-06	13.21	8.12	9.96	6.67	5.37	4.50	10.67	4.42	7.17	7.50	8.12	13.17	6	0

A data.frame: 6 × 15

Question 73

년도 - 월을 기준으로 모든 컬럼의 평균값을 구하여라

```
Ans<-df %>%
    select(-WeekCheck) %>%
    group_by(year=format(df$Yr_Mo_Dy,'%Y-%m')) %>%
    summarise_all("mean", na.rm = TRUE) %>%
    select(-Yr_Mo_Dy)

head(Ans)
```

year	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL	weekday
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1961-01	14.841333	11.988333	13.43161	7.736774	11.072759	8.588065	11.184839	9.245333	9.085806	10.107419	13.88097	14.70323	3.806452
1961-02	16.269286	14.975357	14.44148	9.230741	13.852143	10.937500	11.890714	11.846071	11.821429	12.714286	18.58321	15.41179	4.000000
1961-03	10.890000	11.296452	10.75290	7.284000	10.509355	8.866774	9.644194	9.829677	10.294138	11.251935	16.41097	15.72000	4.096774
1961-04	10.722667	9.427667	9.99800	5.830667	8.435000	6.495000	6.925333	7.094667	7.342333	7.237000	11.14733	10.27833	4.000000
1961-05	9.860968	8.850000	10.81806	5.905333	9.490323	6.574839	7.604000	8.177097	8.039355	8.499355	11.90032	12.01161	3.903226
1961-06	9.904138	8.520333	8.86700	6.083000	10.824000	6.707333	9.095667	8.849333	9.086667	9.940333	13.99500	14.55379	4.100000

A tibble: 6 × 14

Question 74

RPT 컬럼의 값을 일자별 기준으로 1차 차분하라

Question 75

RPT와 VAL의 컬럼을 일주일 간격으로 각각 이동평균한값을 구하여라

```
library(zoo)
Ans<-rollmean(df[,c(2:13)], k = 13, fill = NA, align = "center")
head(Ans,10)
```

```
Attaching package: ‘zoo’
```

```
The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric
```

RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
13.54077	NA	11.65077	6.429231	NA	6.609231	10.636154	NA	8.046923	8.981538	13.00308	13.30462
13.07923	NA	11.18231	5.766154	8.492308	5.956154	9.821538	NA	7.252308	8.219231	12.13154	12.54538
12.87385	10.130000	11.25308	5.448462	8.089231	5.606923	9.129231	NA	6.864615	7.908462	11.36231	12.00077
12.71385	9.696923	11.51000	5.031538	8.102308	5.664615	8.975385	6.63	6.842308	7.882308	11.38462	11.71846

A matrix: 10 × 12 of type dbl

Attention

서울시 미세먼지 데이터 : https://www.airkorea.or.kr/web/realSearch?pMENU_NO=97 **DataUrl = ‘https://raw.githubusercontent.com/Datamanim/pandas/main/seoul_pm.csv’**

Question 76

년-월-일:시 컬럼을 dttm 형태로 변경하라. 서울시의 제공데이터의 경우 0시가 24시로 표현된다. 데이터프레임명은 df로 하라

```
df<- read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/seoul_pm.csv')

library("stringr")
a<-as.data.frame(str_split_fixed(df$`X.년.월.일.시.`,':',2))
colnames(a) <- c('date','hour')

df$`X.년.월.일.시.` <-a %>%
    mutate(total = paste(date,' ',hour)) %>%
    mutate(newdate = ifelse(hour==24,
                        paste(as.character(as.Date(date) + days(1)),' ','00'),
                        paste(as.character(as.Date(date)) ,' ',hour)
                           )) %>%
    select(newdate) %>%
    pull(newdate)

df$`X.년.월.일.시.`<- strptime(df$`X.년.월.일.시.`,format='%Y-%m-%d %H')
head(df)
```

	X.년.월.일.시.	PM10 등급	PM10	PM2.5 등급	PM2.5	오존 등급	오존	이산화질소등급	이산화질소	일산화탄소등급	일산화탄소	아황산가스등급	아황산가스
	<dttm>	<chr>	<int>	<chr>	<int>	<chr>	<dbl>	<chr>	<dbl>	<chr>	<dbl>	<chr>	<dbl>
1	2021-05-15 15:00:00	보통	47	보통	19	좋음	0.017	좋음	0.023	좋음	0.4	좋음	0.003
2	2021-05-15 14:00:00	보통	43	보통	20	좋음	0.024	좋음	0.019	좋음	0.3	좋음	0.003
3	2021-05-15 13:00:00	보통	34	보통	24	보통	0.035	좋음	0.017	좋음	0.4	좋음	0.004
4	2021-05-15 12:00:00	보통	41	보통	27	보통	0.037	좋음	0.020	좋음	0.4	좋음	0.004
5	2021-05-15 11:00:00	보통	51	보통	34	보통	0.033	좋음	0.023	좋음	0.4	좋음	0.005
6	2021-05-15 10:00:00	보통	47	보통	31	좋음	0.028	좋음	0.029	좋음	0.5	좋음	0.005

A data.frame: 6 × 13

Question 77

	X.년.월.일.시.	PM10 등급	PM10	PM2.5 등급	PM2.5	오존 등급	오존	이산화질소등급	이산화질소	일산화탄소등급	일산화탄소	아황산가스등급	아황산가스	dayName
	<dttm>	<chr>	<int>	<chr>	<int>	<chr>	<dbl>	<chr>	<dbl>	<chr>	<dbl>	<chr>	<dbl>	<chr>
1	2021-05-15 15:00:00	보통	47	보통	19	좋음	0.017	좋음	0.023	좋음	0.4	좋음	0.003	Saturday
2	2021-05-15 14:00:00	보통	43	보통	20	좋음	0.024	좋음	0.019	좋음	0.3	좋음	0.003	Saturday
3	2021-05-15 13:00:00	보통	34	보통	24	보통	0.035	좋음	0.017	좋음	0.4	좋음	0.004	Saturday
4	2021-05-15 12:00:00	보통	41	보통	27	보통	0.037	좋음	0.020	좋음	0.4	좋음	0.004	Saturday
5	2021-05-15 11:00:00	보통	51	보통	34	보통	0.033	좋음	0.023	좋음	0.4	좋음	0.005	Saturday
6	2021-05-15 10:00:00	보통	47	보통	31	좋음	0.028	좋음	0.029	좋음	0.5	좋음	0.005	Saturday

A data.frame: 6 × 14

Question 78

일자별 각 PM10등급의 빈도수를 파악하라

```
Ans<-df %>%
    group_by(dayName) %>%
    count(PM10등급)
head(Ans)
```

dayName	PM10등급	n
<chr>	<chr>	<int>
Friday		3
Friday	나쁨	31
Friday	매우나쁨	17
Friday	보통	120
Friday	좋음	21
Monday	나쁨	1

A grouped_df: 6 × 3

Question 79

시간이 연속적으로 존재하며 결측치가 없는지 확인하라

```
sum(diff(df$`X.년.월.일.시.`) !=-1)
```

0

Question 80

오전 10시와 오후 10시(22시)의 PM10의 평균값을 각각 구하여라

```
Ans<-df %>%
    filter(hour(`X.년.월.일.시.`)== 10 | hour(`X.년.월.일.시.`)== 22) %>%
    group_by(hour=hour(`X.년.월.일.시.`)) %>%
    summarize(mean = mean(PM10))
Ans
```

hour	mean
<int>	<dbl>
10	70.38462
22	69.94118

A tibble: 2 × 2

Question 81

날짜 컬럼을 index로 만들어서 df2에 저장하라

```
df2<-df
rownames(df2) <-df2$`X.년.월.일.시.`
df2<-df2[,c(2:14)]
head(df2)
```

	PM10 등급	PM10	PM2.5 등급	PM2.5	오존 등급	오존	이산화질소등급	이산화질소	일산화탄소등급	일산화탄소	아황산가스등급	아황산가스	dayName
	<chr>	<int>	<chr>	<int>	<chr>	<dbl>	<chr>	<dbl>	<chr>	<dbl>	<chr>	<dbl>	<chr>
2021-05-15 15:00:00	보통	47	보통	19	좋음	0.017	좋음	0.023	좋음	0.4	좋음	0.003	Saturday
2021-05-15 14:00:00	보통	43	보통	20	좋음	0.024	좋음	0.019	좋음	0.3	좋음	0.003	Saturday
2021-05-15 13:00:00	보통	34	보통	24	보통	0.035	좋음	0.017	좋음	0.4	좋음	0.004	Saturday
2021-05-15 12:00:00	보통	41	보통	27	보통	0.037	좋음	0.020	좋음	0.4	좋음	0.004	Saturday
2021-05-15 11:00:00	보통	51	보통	34	보통	0.033	좋음	0.023	좋음	0.4	좋음	0.005	Saturday
2021-05-15 10:00:00	보통	47	보통	31	좋음	0.028	좋음	0.029	좋음	0.5	좋음	0.005	Saturday

A data.frame: 6 × 13

Question 82

df 데이터를 주단위로 뽑아서 최소,최대 평균, 표준표차를 구하여라

```
Ans<-df %>%
    mutate(week= week(df$`X.년.월.일.시.`)) %>%
    select_if(is.numeric) %>%
    group_by(week) %>%
    summarise_all(mean)

Ans
```

week	PM10	PM2.5	오존	이산화질소	일산화탄소	아황산가스
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
12	72.17391	37.69565	0.03517391	0.04017391	0.4956522	0.003521739
13	113.64286	NA	0.02308929	0.03774405	0.5642857	0.003017857
14	38.24405	17.74405	0.03406548	0.02331548	0.3886905	0.002672619
15	37.45833	19.48214	0.03642262	0.02322024	0.3827381	0.002619048
16	NA	NA	NA	NA	NA	NA
17	NA	17.93452	0.03705952	0.02217857	0.3928571	0.002702381
18	NA	NA	NA	NA	NA	NA
19	NA	NA	NA	NA	NA	NA
20	63.52500	38.10000	0.03235000	0.03285000	0.5000000	0.004500000

A tibble: 9 × 7

06_Reshape(Pivot)

Attention

국가별 5세이하 사망비율 통계 : https://www.kaggle.com/utkarshxy/who-worldhealth-statistics-2020-complete Dataurl = 'https://raw.githubusercontent.com/Datamanim/pandas/main/under5MortalityRate.csv'

Question 83

Indicator을 삭제하고 First Tooltip 컬럼에서 신뢰구간에 해당하는 표현([~~])을 지우고 first 컬럼에 실수형으로 타입을 변경한 후 추가하라

```
df<-
read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/under5MortalityRate.csv'
)
```

```
library(dplyr)
library(stringr)

df<- df %>%
    select(-c(Indicator))  %>%
    mutate(first = as.numeric(sapply(strsplit(df$First.Tooltip, "\\["), head, 1))) %>%
    select(-c(First.Tooltip))

head(df)
```

	Location	Period	Dim1	first
	<chr>	<int>	<chr>	<dbl>
1	Afghanistan	2019	Both sexes	60.27
2	Afghanistan	2019	Male	63.83
3	Afghanistan	2019	Female	56.57
4	Afghanistan	2018	Both sexes	62.54
5	Afghanistan	2018	Male	66.08
6	Afghanistan	2018	Female	58.84

A data.frame: 6 × 4

Question 84

년도가 2015년 이상, Dim1이 Both sexes인 케이스만 추출하라

```
df2<-df %>%
    filter(Period >=2015 & Dim1 =='Both sexes')
head(df2)
```

	Location	Period	Dim1	first
	<chr>	<int>	<chr>	<dbl>
1	Afghanistan	2019	Both sexes	60.27
2	Afghanistan	2018	Both sexes	62.54
3	Afghanistan	2017	Both sexes	64.94
4	Afghanistan	2016	Both sexes	67.57
5	Afghanistan	2015	Both sexes	70.44
6	Albania	2019	Both sexes	9.68

A data.frame: 6 × 4

Question 85

84번 문제에서 추출한 데이터로 아래와 같이 나라에 따른 년도별 사망률을 데이터 프레임화 하라

```
library(reshape2)
Ans<-df2 %>%
    select(-c(Dim1)) %>%
    group_by(Location,first) %>%
    dcast(Location~Period,value.var='first')
head(Ans)
```

```
Attaching package: ‘reshape2’
```

```
The following object is masked from ‘package:tidyr’:

    smiths
```

```
The following objects are masked from ‘package:reshape’:

    colsplit, melt, recast
```

	Location	2015	2016	2017	2018	2019
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Afghanistan	70.44	67.57	64.94	62.54	60.27
2	Albania	9.57	9.42	9.42	9.53	9.68
3	Algeria	25.18	24.79	24.32	23.81	23.26
4	Andorra	3.53	3.37	3.22	3.09	2.97
5	Angola	88.20	84.21	80.62	77.67	74.69
6	Antigua and Barbuda	7.75	7.42	7.12	6.85	6.61

A data.frame: 6 × 6

Question 86

전체 데이터(df)에서 Dim1에 따른 년도별 사망비율의 평균을 구하라

```
Ans<-df %>%
    select(-c(Location)) %>%
    group_by(Dim1,Period) %>%
    summarise_all(mean) %>%
    ungroup() %>%
    dcast(Period~Dim1,value.var='first')

head(Ans)
```

	Period	Both sexes	Female	Male
	<int>	<dbl>	<dbl>	<dbl>
1	1950	147.7008	140.9098	154.1512
2	1951	155.5375	149.2102	161.5382
3	1952	157.8111	151.5161	163.7608
4	1953	156.1472	150.2509	161.7421
5	1954	154.5399	148.6883	160.0810
6	1955	155.7972	149.8432	161.4569

A data.frame: 6 × 4

Attention

올림픽 메달리스트 정보 데이터: https://www.kaggle.com/the-guardian/olympic-games dataUrl ='https://raw.githubusercontent.com/Datamanim/pandas/main/winter.csv'

Question 87

데이터에서 Country가 한국(KOR) 데이터만 추출하라

```
df <-read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/winter.csv')

kor <- df %>%
        filter(Country =='KOR')
head(kor)
```

	Year	City	Sport	Discipline	Athlete	Country	Gender	Event	Medal
	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	1992	Albertville	Skating	Short Track Speed Skating	LEE, Jun-Ho	KOR	Men	1000M	Bronze
2	1992	Albertville	Skating	Short Track Speed Skating	KIM, Ki-Hoon	KOR	Men	1000M	Gold
3	1992	Albertville	Skating	Short Track Speed Skating	KIM, Ki-Hoon	KOR	Men	5000M Relay	Gold
4	1992	Albertville	Skating	Short Track Speed Skating	LEE, Jun-Ho	KOR	Men	5000M Relay	Gold
5	1992	Albertville	Skating	Short Track Speed Skating	MO, Ji-Soo	KOR	Men	5000M Relay	Gold
6	1992	Albertville	Skating	Short Track Speed Skating	SONG, Jae-Kun	KOR	Men	5000M Relay	Gold

A data.frame: 6 × 9

Question 88

한국 올림픽 메달리스트 데이터에서 년도에 따른 medal 종류별 갯수를 데이터프레임화 하라

```
Ans<-kor %>%
    group_by(Year,Medal) %>%
    count() %>%
    ungroup() %>%
    dcast(Year~Medal,value.var='n')
Ans
```

Year	Bronze	Gold	Silver
<int>	<int>	<int>	<int>
1992	1	5	1
1994	1	8	1
1998	2	6	4
2002	NA	5	2
2006	2	14	3
2010	2	6	10
2014	2	7	5

A data.frame: 7 × 4

Question 89

전체 데이터에서 sport종류에 따른 성별수를 구하여라

```
Ans<-df %>%
    group_by(Sport,Gender) %>%
    count() %>%
    dcast(Sport~Gender,value.var='n')

Ans
```

Sport	Men	Women
<chr>	<int>	<int>
Biathlon	270	150
Bobsleigh	416	36
Curling	97	75
Ice Hockey	1231	305
Luge	135	45
Skating	665	564
Skiing	1130	651

A data.frame: 7 × 3

Question 90

전체 데이터에서 Discipline종류에 따른 따른 Medal수를 구하여라

```
Ans<-df %>%
    group_by(Discipline,Medal) %>%
    count() %>%
    dcast(Discipline~Medal,value.var='n')


head(Ans)
```

	Discipline	Bronze	Gold	Silver
	<chr>	<int>	<int>	<int>
1	Alpine Skiing	141	143	144
2	Biathlon	139	140	141
3	Bobsleigh	147	134	141
4	Cross Country Skiing	263	264	262
5	Curling	56	58	58
6	Figure skating	118	122	119

A data.frame: 6 × 4

07_Merge , Concat

Attention

국가별 5세이하 사망비율 통계 : https://www.kaggle.com/utkarshxy/who-worldhealth-statistics-2020-complete
데이터 변형
Dataurl = 'https://raw.githubusercontent.com/Datamanim/pandas/main/mergeTEst.csv'

Question 91

df1과 df2 데이터를 하나의 데이터 프레임으로 합쳐라

```
df<-
read.csv('https://raw.githubusercontent.com/Datamanim/pandas/main/mergeTEst.csv',row.names = 1)

df1<-df[c(1,2,3,4),]
df2<-df[c(5:nrow(df)),]
```

```
Ans<-rbind(df1,df2)
head(Ans)
```

	X2010	X2011	X2012	X2013	X2014	X2015	X2016	X2017	X2018	X2019
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Afghanistan	64.023	61.640	59.367	57.170	55.08	53.107	51.267	49.560	47.983	46.453
Albania	11.803	10.807	9.943	9.267	8.79	8.493	8.363	8.363	8.453	8.597
Algeria	23.540	22.907	22.450	22.117	21.85	21.587	21.257	20.850	20.407	19.930
Andorra	4.240	4.033	3.843	3.667	3.49	3.330	3.187	3.060	2.933	2.827
Angola	75.713	71.280	67.233	63.570	60.43	57.757	55.510	53.460	51.757	50.093
Antigua and Barbuda	8.667	8.223	7.807	7.420	7.07	6.757	6.483	6.230	6.000	5.783

A data.frame: 6 × 10

Question 92

df3과 df4 데이터를 하나의 데이터 프레임으로 합쳐라. 둘다 포함하고 있는 년도에 대해서만 고려한다

	X2010	X2011	X2012	X2013
	<dbl>	<dbl>	<dbl>	<dbl>
Afghanistan	64.023	61.640	59.367	57.170
Albania	11.803	10.807	9.943	9.267
Algeria	23.540	22.907	22.450	22.117

A data.frame: 3 × 4

```
df4
```

	X2012	X2013	X2014	X2015	X2016	X2017	X2018	X2019
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Angola	67.233	63.570	60.430	57.757	55.510	53.460	51.757	50.093
Antigua and Barbuda	7.807	7.420	7.070	6.757	6.483	6.230	6.000	5.783
Argentina	11.840	11.283	10.733	10.203	9.683	9.177	8.680	8.227
Armenia	14.897	14.170	13.477	12.817	12.183	11.583	11.007	10.497
Australia	3.623	3.467	3.343	3.253	3.183	3.137	3.090	3.047
Austria	3.333	3.210	3.113	3.043	2.987	2.943	2.897	2.843

A data.frame: 6 × 8

Question 93

df3과 df4 데이터를 하나의 데이터 프레임으로 합쳐라. 모든 컬럼을 포함하고, 결측치는 0으로 대체한다

Question 94

df5과 df6 데이터를 하나의 데이터 프레임으로 merge함수를 이용하여 합쳐라. Algeria컬럼을 key로 하고 두 데이터 모두 포함하는 데이터만 출력하라

```
df5<-t(df)[c(1:7),c(1:3)]
df6<-t(df)[c(6:nrow(t(df))),c(2,3,4,5)]
```

Question 95

df5과 df6 데이터를 하나의 데이터 프레임으로 merge함수를 이용하여 합쳐라. Algeria컬럼을 key로 하고 합집합으로 합쳐라

머지 잘모르겠어요.. 답아시면 깃헙에 코드 공유 부탁드려요 ㅠㅠ