



HADOOP 을 이용한  
**육류 소비구조의  
분석 및 시장 예측**

# 목차



배경



데이터 수집 및 전처리



데이터 분석 및 결과



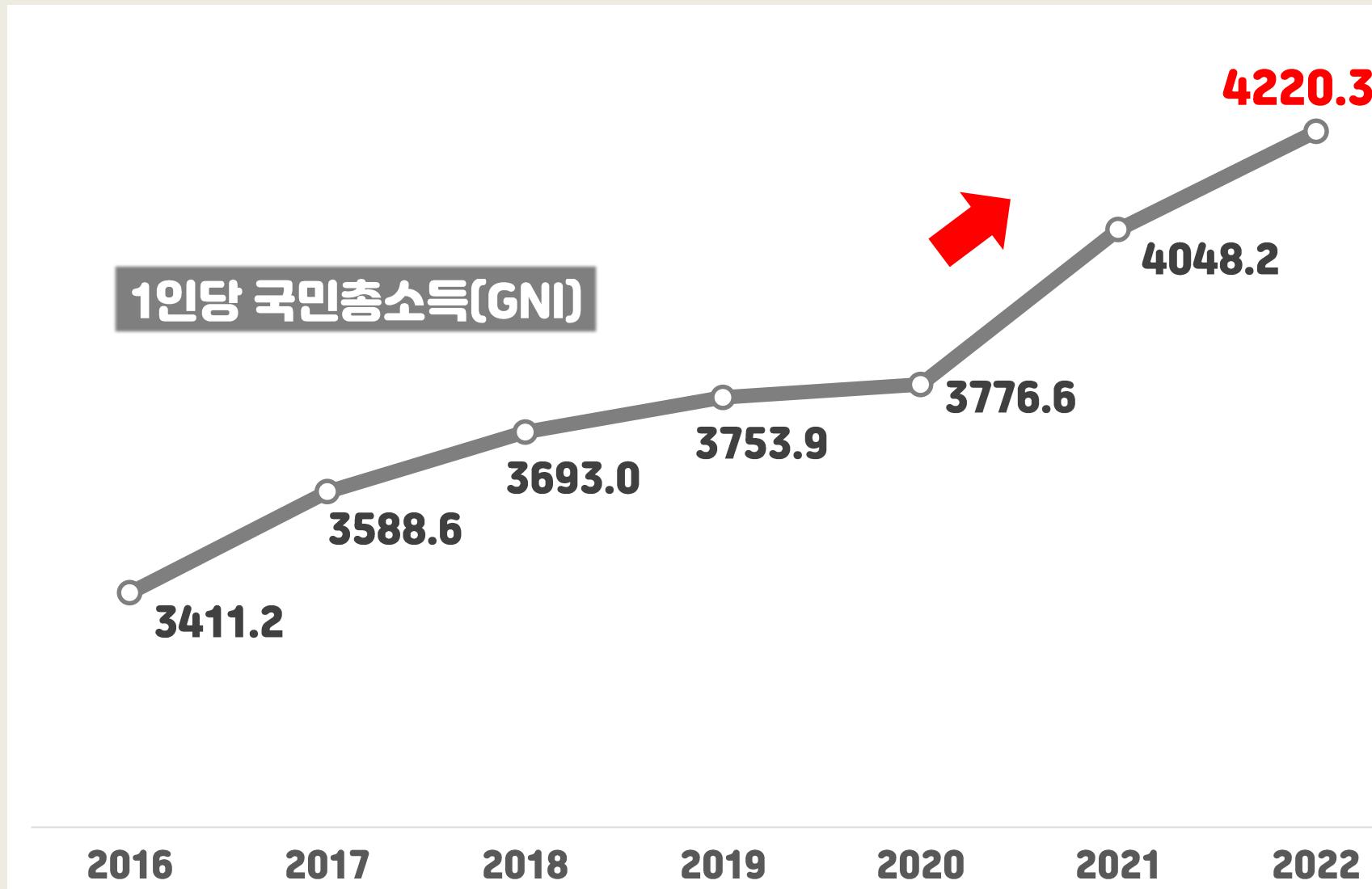
대안 및 기대효과



데이터 목록 및 출처 / 참고문헌

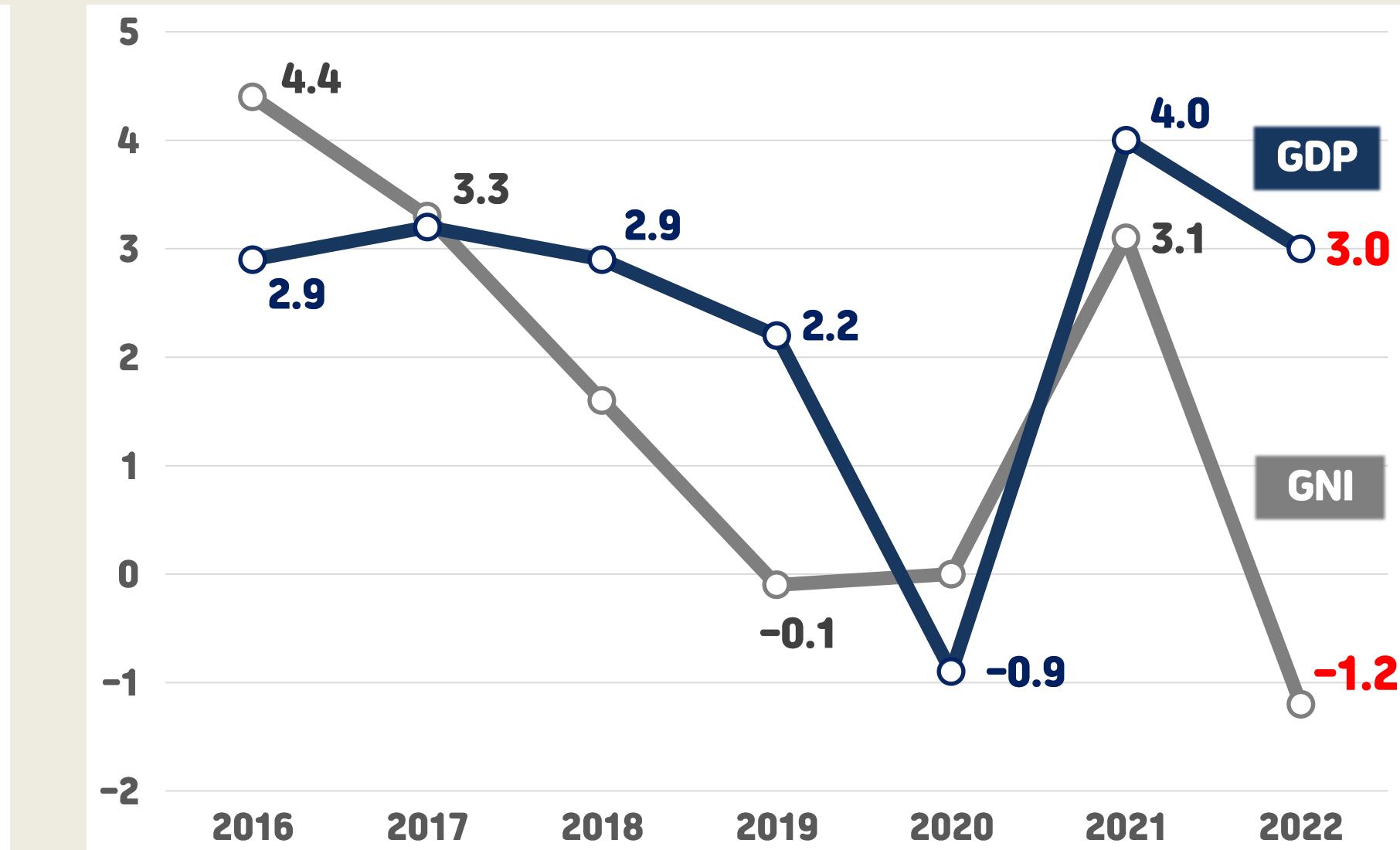
# 1. 사회 · 경제적 변화

**1인당 국민총소득(GNI)**



1인당 국민총소득(GNI) 계속 증가

**경제성장률 (%)**



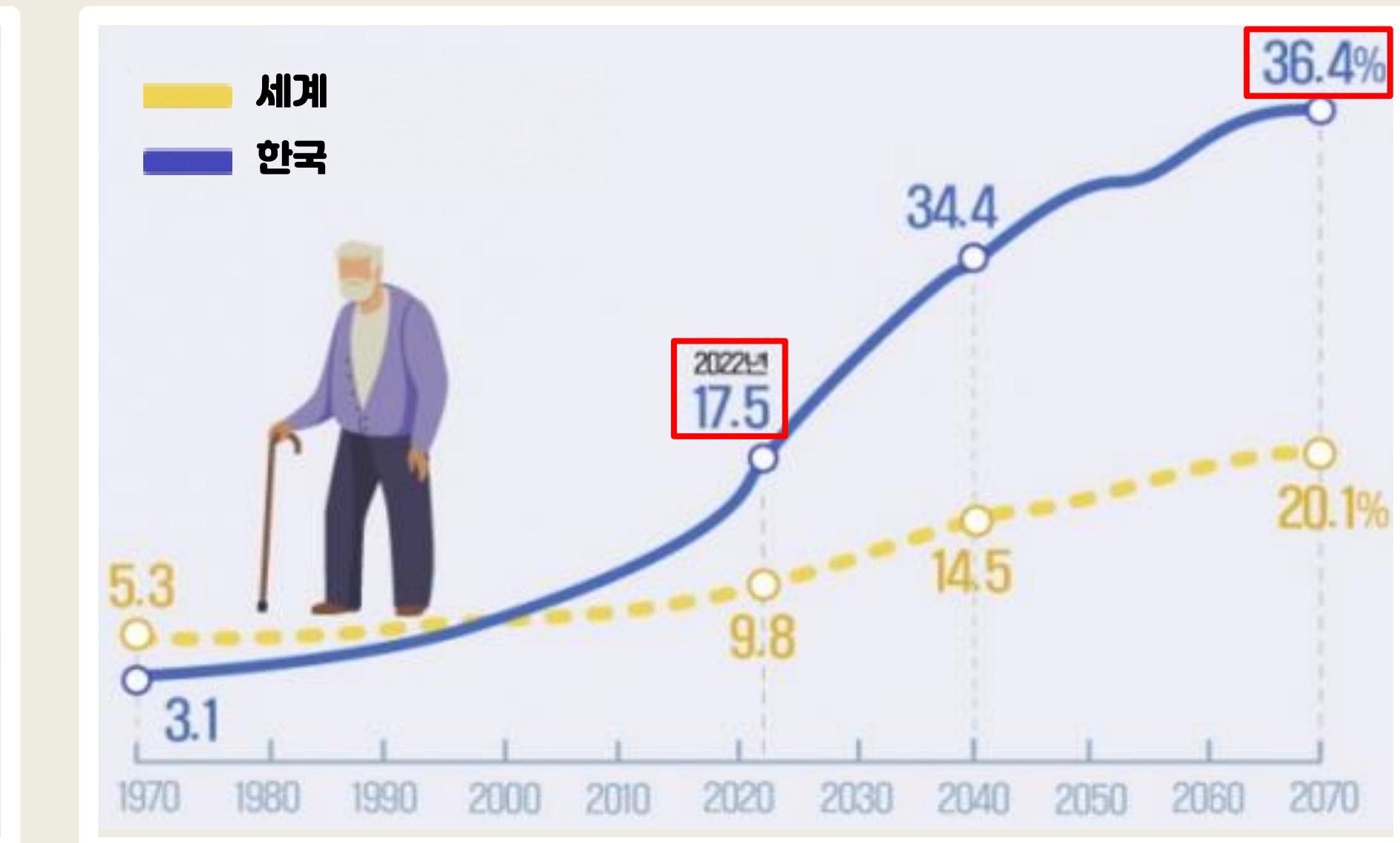
경제성장률 2021년 이후 하락세  
\*\* 2019 : 코로나, 2022 : 러·우 전쟁 영향

# 1. 사회 · 경제적 변화

## 인구 성장률



## 고령인구 비중

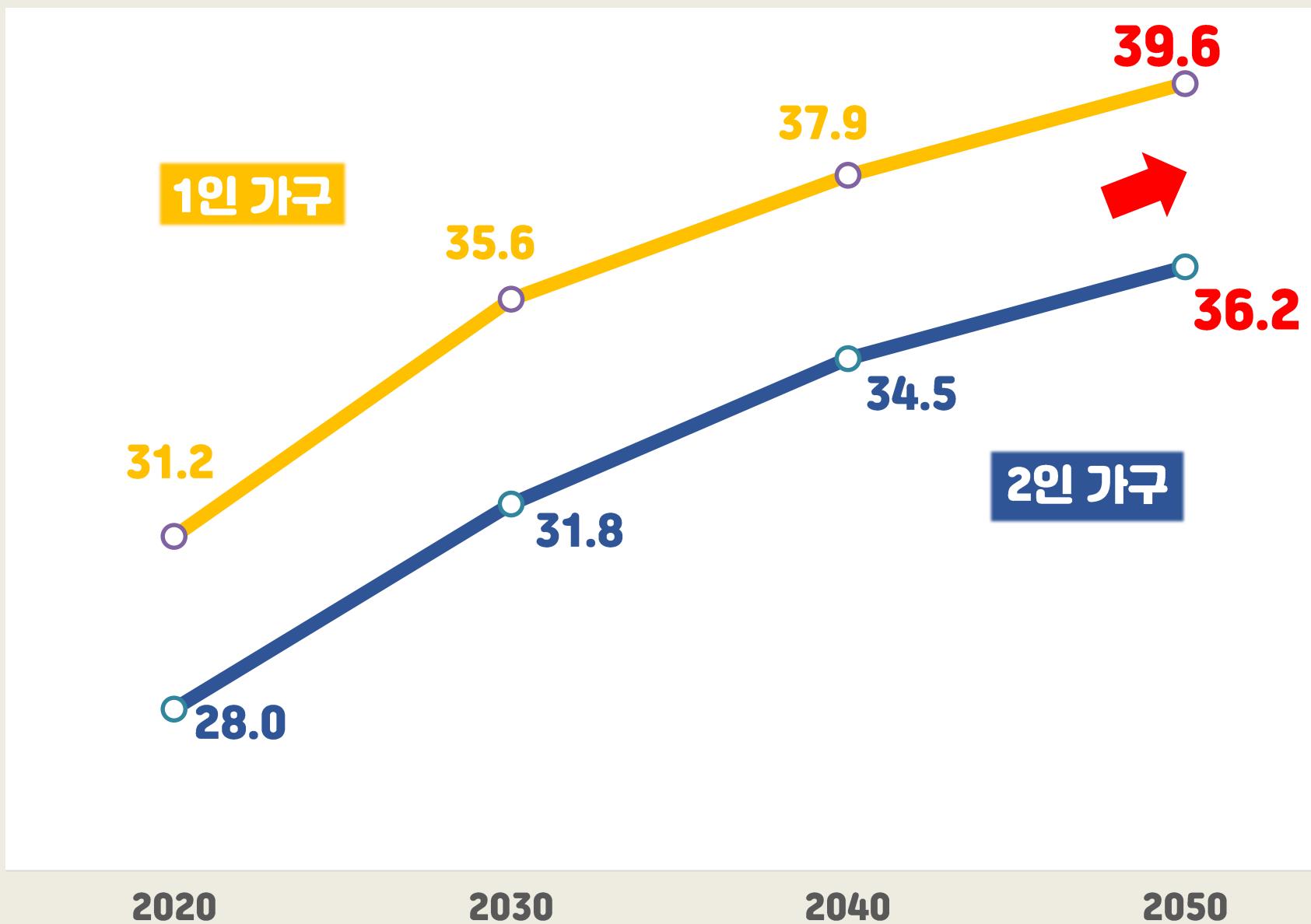


세계 인구성장률에 비해 우리나라의 인구는 매우 낮은 편  
▷ 계속 감소 추세로 예상

고령인구의 증가 가속화 예상

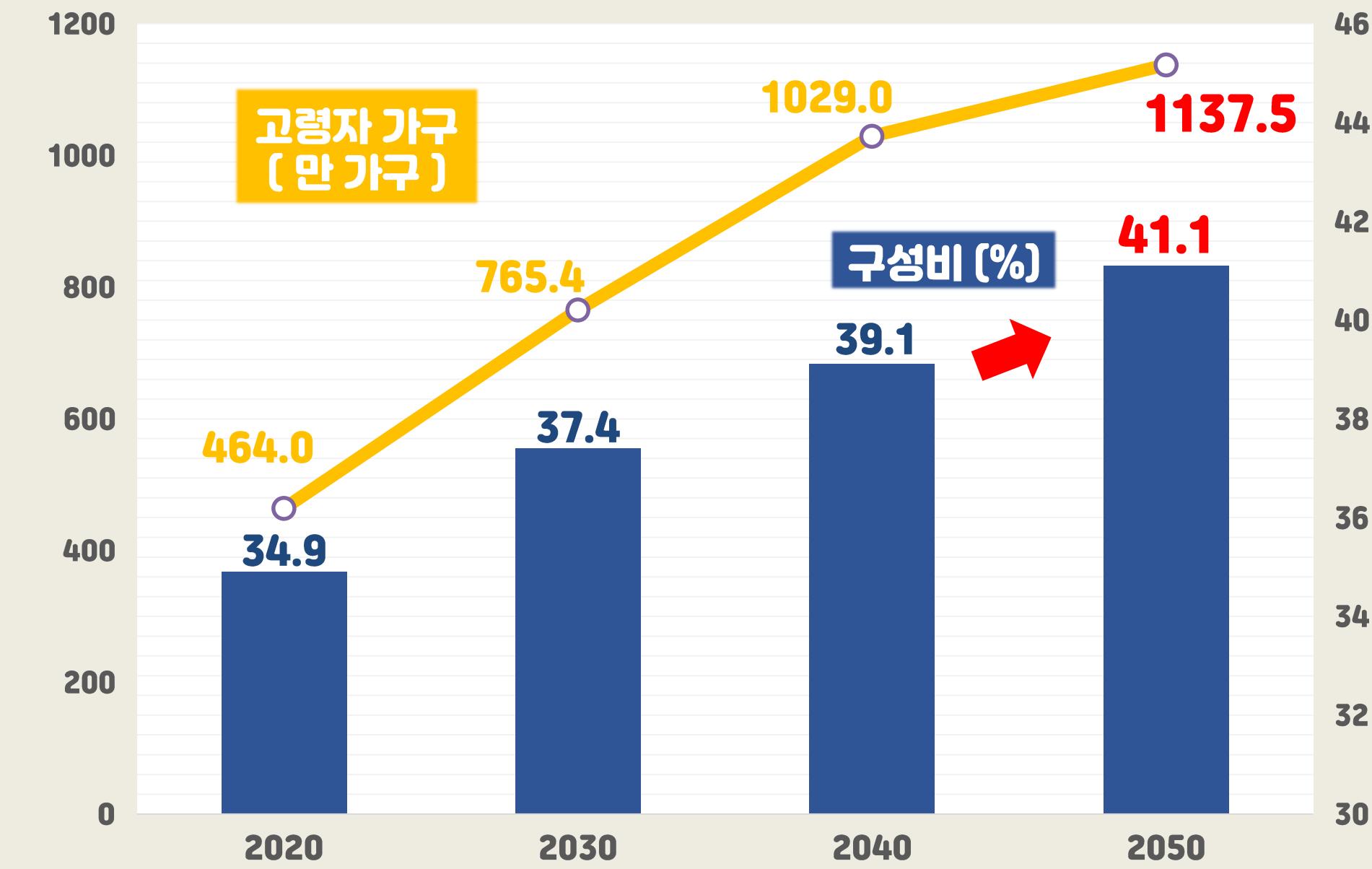
# 1. 사회 · 경제적 변화

## 1 · 2인 가구 증가 추세



1 · 2인 가구 가 점점 더 증가 예상

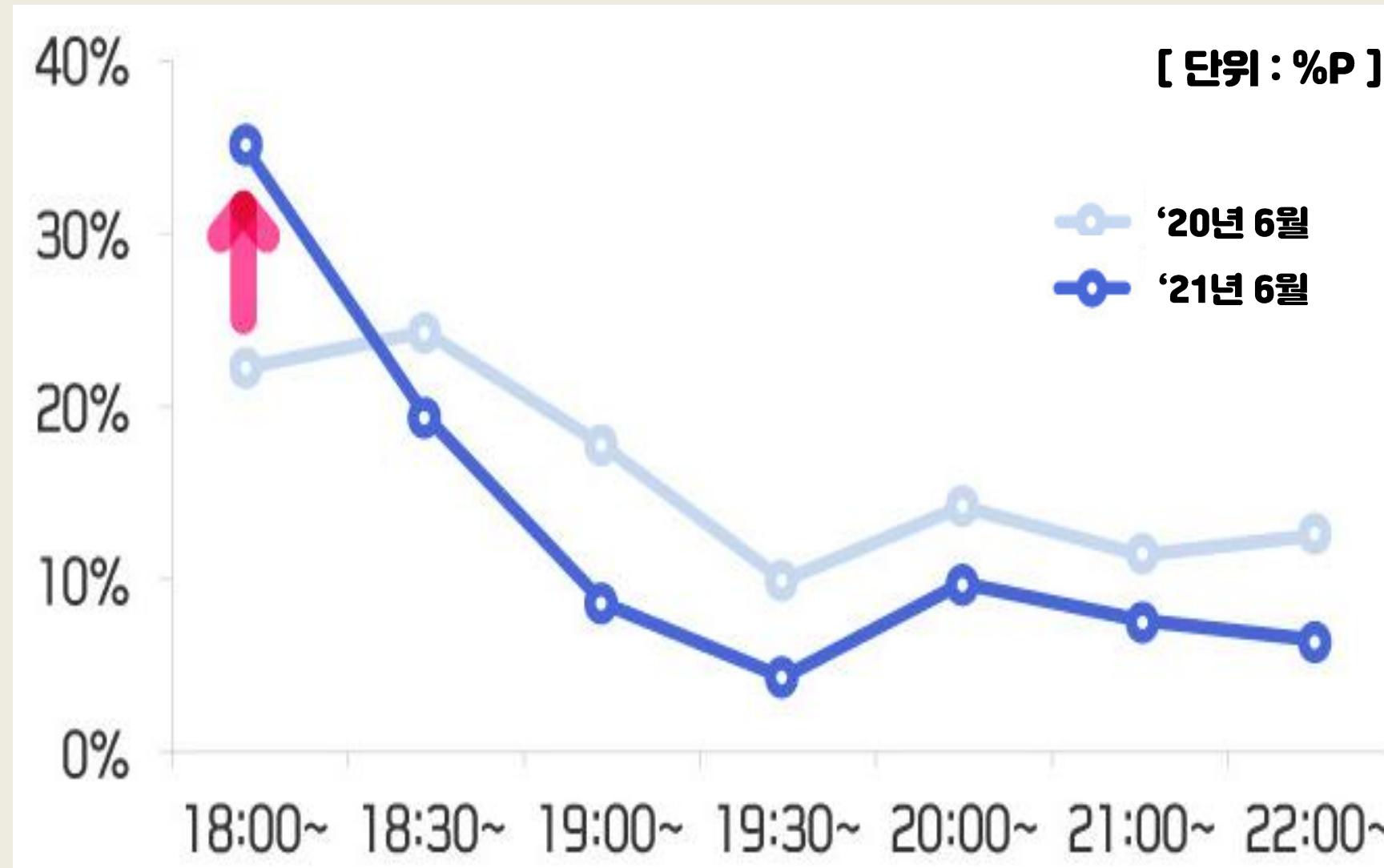
## 고령자 가구 추세 및 구성비



고령자 가구 도 더욱 증가 예상

## 2. 제도적 변화 – 주 52시간제 시행

**퇴근 시간 비중 변화**



직장인들의 퇴근시간 비중이 18시가 압도적으로 높음

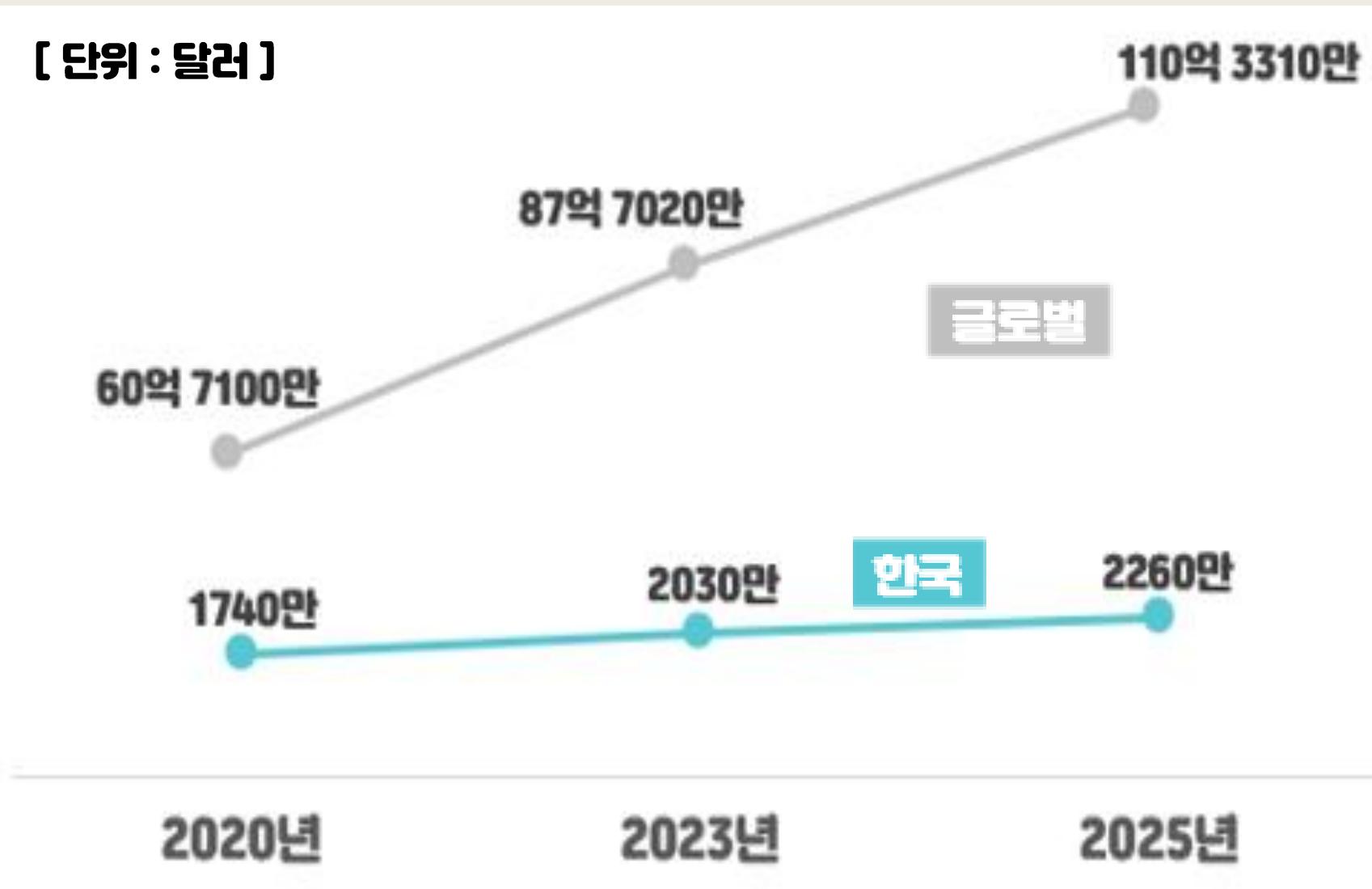
**법인카드 이용시간 비중 변화**



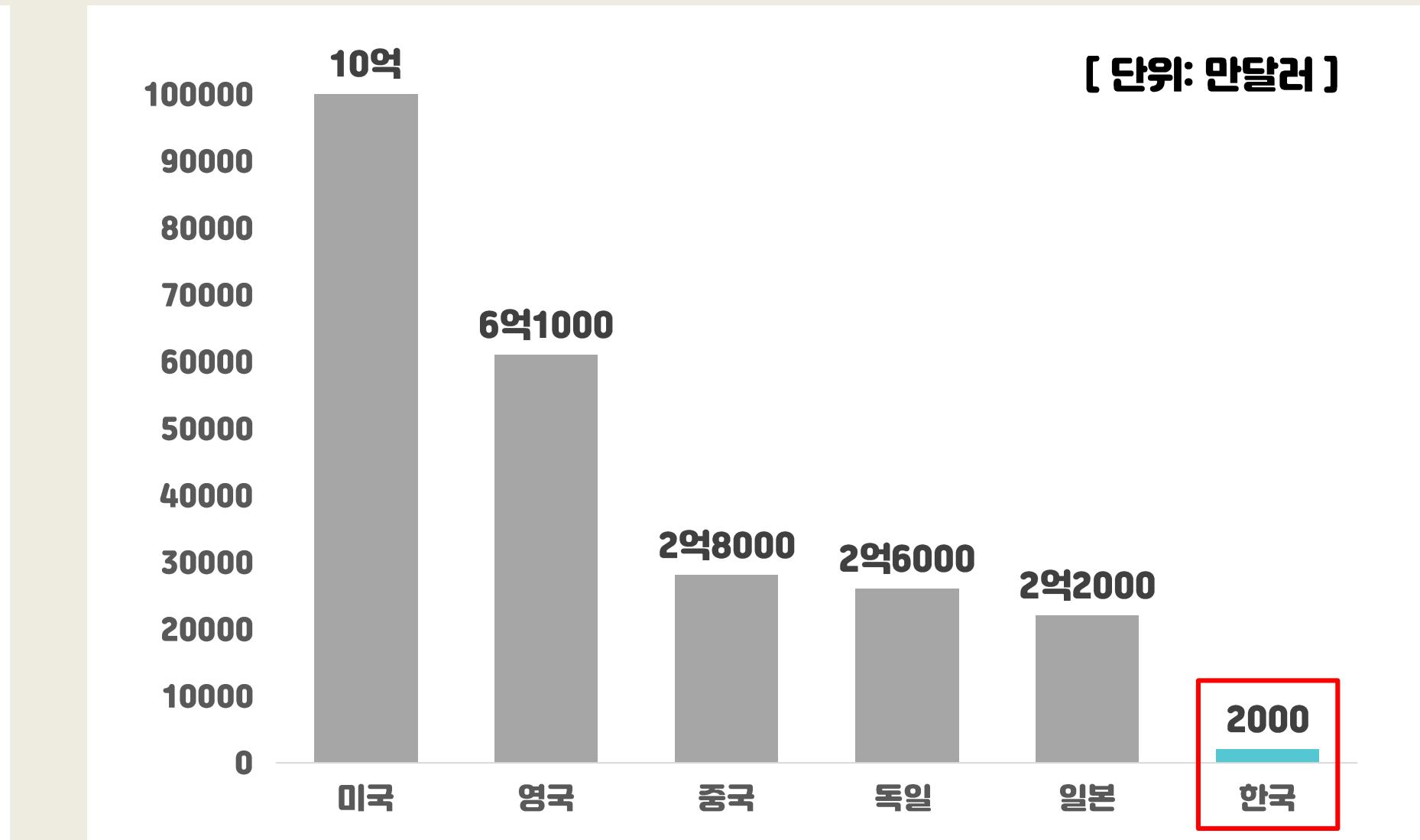
점심시간에 하는 회식이 증가

### 3. 과학 기술적 변화

#### 국내 식물성 대체육 시장 규모



#### 글로벌 대체육 시장 규모



국내 식물성 대체육 시장은  
글로벌 시장에 비해 상당히 작은 규모

국가별 대체육 시장에서도 한국은 상당히 적은 수준

## ◆ HADOOP-HIVE 연동

### # NAMENODE Format

```
[root@namenode local]# hadoop namenode -format
23/04/05 16:58:06 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = namenode/192.168.111.130
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 1.0.4
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0
-r 1393290; compiled by 'hortonfo' on Wed Oct 3 05:13:58 UTC 2012
*****
23/04/05 16:58:06 INFO util.GSet: VM type      = 64-bit
23/04/05 16:58:06 INFO util.GSet: 2%max memory = 19.33375 MB
23/04/05 16:58:06 INFO util.GSet: capacity      = 2^21 = 2097152 entries
23/04/05 16:58:06 INFO util.GSet: recommended=2097152, actual=2097152
23/04/05 16:58:07 INFO namenode.FSNamesystem: fsOwner=root
23/04/05 16:58:07 INFO namenode.FSNamesystem: supergroup=supergroup
23/04/05 16:58:07 INFO namenode.FSNamesystem: isPermissionEnabled=true
23/04/05 16:58:07 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
23/04/05 16:58:07 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessTokenUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
23/04/05 16:58:07 INFO namenode.NameNode: Caching file names occurring more than 10 times
23/04/05 16:58:07 INFO common.Storage: Image file of size 110 saved in 0 seconds.
23/04/05 16:58:07 INFO common.Storage: Storage directory /home/hadoop/hdfs/temp/dfs/name has been successfully formatted.
23/04/05 16:58:07 INFO namenode.NameNode: SHUTDOWN_MSG:
```

### # HADOOP Cluster start

```
[root@namenode local]# start-all.sh
starting namenode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hanyoung-namenode-namenode.out
datanode1: datanode running as process 114736. Stop it first.
snamenode: datanode running as process 124487. Stop it first.
datanode2: ssh: connect to host datanode2 port 22: No route to host
snamenode: secondarynamenode running as process 62284. Stop it first.
jobtracker: running as process 5220. Stop it first.
datanode1: tasktracker running as process 51729. Stop it first.
snamenode: tasktracker running as process 62403. Stop it first.
datanode2: ssh: connect to host datanode2 port 22: No route to host
```

```
[root@namenode local]# sps
[root@namenode local]# mr
[root@namenode local]# tr
[root@namenode local]# ips
```

디렉토리 내 파일 삭제 /  
방화벽 제거 / ip 주소 확정

### # Jps : HADOOP Cluster 실행 확인

#### NAMENODE

```
[root@namenode conf]# jps
43603 JobTracker
43428 NameNode
43718 Jps
```

#### SNAMENODE

```
[root@snamenode local]# jps
11122 Jps
62403 TaskTracker
124487 DataNode
62284 SecondaryNameNode
```

#### DATANODE1

```
[root@datanode1 local]# jps
51729 TaskTracker
114736 DataNode
115621 Jps
```

#### DATANODE2

```
[root@datanode2 local]# jps
16304 DataNode
81845 TaskTracker
17198 Jps
```

## ◆ HIVE – 변수 설정

### # 변수 설정

연도 - year
가구원수 - cnt
취업인원수 - hiredcnt
노인가구여부 - eldercnt
맞벌이가구여부 - dualincomecnt
일반가구여부 - normalcnt
성별코드 - gendercode
연령 - age
학력코드 - educode
학력 - edu
주거용전용면적 - residentialarea
주택소유유무 - housecnt
주택소유구분 - housecode
가계소비지출금액 - spending
식대 - meals
육류구입비 - meat
육류가공품구입비 - meatproducts



### # 2014.CSV

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Year	Cnt	Hiredcnt	ElderCnt	DualincomeCnt	NormalCnt	GenderCode	Age	EduCode	Edu	ResidentialArea	HouseCnt	HouseCode	Spending	Meals	Meat	MeatProducts
2014	4	1	0	0	1	1	44	3	고등학교	47	1	YES	2086494	153190	46725	20172
2014	4	1	0	0	1	1	44	5	대학교(4년)	89	2	NO	2620934	376911	29974	12691
2014	1	0	1	0	0	2	79	0	무학	33	1	YES	819943	1183	118285	0
2014	1	0	1	0	0	1	66	1	초등학교	26	1	YES	3239180	35000	10000	0
2014	3	1	0	0	1	1	60	1	초등학교	79	2	NO	1330790	161700	102000	0
2014	1	1	1	0	0	2	74	1	초등학교	79	2	NO	1788962	20600	29793	0
2014	4	2	0	1	0	1	45	5	대학교(4년)	85	1	YES	3494897	636076	73410	12407
2014	4	1	0	0	1	1	37	5	대학교(4년)	84	1	YES	3459741	255358	66097	23046
2014	2	1	0	0	1	1	58	3	고등학교	84	1	YES	7018826	823499	6459	4910
2014	3	1	0	0	1	1	60	1	초등학교	26	2	NO	3637783	572463	49970	5894
2014	3	2	0	1	0	1	60	3	고등학교	83	1	YES	1725866	209405	34325	9168
2014	4	2	0	1	0	1	36	3	고등학교	60	1	YES	2817744	550357	29741	8105
2014	4	1	0	0	1	1	46	5	대학교(4년)	78	1	YES	3630678	453551	154163	51987
2014	3	2	0	0	1	1	58	3	고등학교	78	1	YES	2435549	861921	16884	3337
2014	1	0	1	0	0	2	73	0	무학	26	2	NO	391844	6730	18865	3753
2014	4	2	0	1	0	1	46	5	대학교(4년)	85	1	YES	6287207	540000	170000	30000
2014	4	1	0	0	1	1	41	4	전문대(초급)	60	1	YES	3969530	459381	48047	28379
2014	2	1	0	0	1	1	68	3	고등학교	50	1	YES	1621547	66943	57494	0
2014	3	1	0	0	1	1	35	5	대학교(4년)	115	1	YES	2606430	389041	24779	6838
2014	4	1	0	0	1	1	36	4	전문대(초급)	76	1	YES	1653050	245089	47974	7619
2014	3	2	0	1	0	1	49	3	고등학교	50	2	NO	1427571	300071	52302	7494
2014	1	1	0	0	1	1	52	3	고등학교	50	1	YES	1431840	286075	94115	2485
2014	4	1	0	0	1	1	43	5	대학교(4년)	116	1	YES	2772114	188209	94570	348
2014	5	2	0	1	0	1	36	4	전문대(초급)	75	1	YES	3463950	139009	31615	17983

## ◆ HIVE – TABLE 구성 / Data Load

### # TABLE 구성

```
CREATE TABLE meat(
    Year INT,
    Cnt INT,
    HiredCnt INT,
    ElderCnt INT,
    DualincomeCnt INT,
    NormalCnt INT,
    GenderCode INT,
    Age INT,
    EduCode INT,
    Edu STRING,
    ResidentialArea INT,
    HouseCnt INT,
    HouseCode STRING,
    Spending INT,
    Meals INT,
    Meat INT,
    MeatProducts INT)
Partitioned by (MeatYear INT)
ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```



### # 년도별 Data Load (2014 ~ 2022)

```
hive> load data local inpath '2014.csv'
      > overwrite into table meat
      > partition(MeatYear=2014);
```

Copying data from file: /mnt/hgfs/Tmp/Data/2014.csv

Copying file: file:/mnt/hgfs/Tmp/Data/2014.csv

Loading data to table default.meat partition (meatyear=2014)

Deleted hdfs://namenode:9000/user/hive/warehouse/meat/meatyear=2014

OK

Time taken: 2.226 seconds

```
hive> show tables;
```

OK

meat

Time taken: 5.346 seconds

```
hive> !ls;
```

2014.csv

2015.csv

2016.csv

2017.csv

2018.csv

2019.csv

2020.csv

2021.csv

2022.csv

**테이블 구성 확인**

**로드된 데이터 확인**

## ◆ HIVE – Warehouse 저장소 확인

### # Warehouse – Load Data 확인

```
[root@namenode Data] # hadoop dfs -lsr /  
drwxr-xr-x  - root supergroup          0 2023-04-04 09:22 /user/hive/warehouse/meat  
drwxr-xr-x  - root supergroup          0 2023-04-04 09:22 /user/hive/warehouse/meat/meatyear=2014  
-rw-r--r--  3 root supergroup 712202 2023-04-04 09:22 /user/hive/warehouse/meat/meatyear=2014/2014.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:39 /user/hive/warehouse/meat/meatyear=2015  
-rw-r--r--  3 root supergroup 707349 2023-04-03 12:39 /user/hive/warehouse/meat/meatyear=2015/2015.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:40 /user/hive/warehouse/meat/meatyear=2016  
-rw-r--r--  3 root supergroup 640016 2023-04-03 12:40 /user/hive/warehouse/meat/meatyear=2016/2016.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:40 /user/hive/warehouse/meat/meatyear=2017  
-rw-r--r--  3 root supergroup 762566 2023-04-03 12:40 /user/hive/warehouse/meat/meatyear=2017/2017.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:41 /user/hive/warehouse/meat/meatyear=2018  
-rw-r--r--  3 root supergroup 757724 2023-04-03 12:41 /user/hive/warehouse/meat/meatyear=2018/2018.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:41 /user/hive/warehouse/meat/meatyear=2019  
-rw-r--r--  3 root supergroup 655322 2023-04-03 12:41 /user/hive/warehouse/meat/meatyear=2019/2019.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:42 /user/hive/warehouse/meat/meatyear=2020  
-rw-r--r--  3 root supergroup 841950 2023-04-03 12:42 /user/hive/warehouse/meat/meatyear=2020/2020.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:42 /user/hive/warehouse/meat/meatyear=2021  
-rw-r--r--  3 root supergroup 864822 2023-04-03 12:42 /user/hive/warehouse/meat/meatyear=2021/2021.csv  
drwxr-xr-x  - root supergroup          0 2023-04-03 12:43 /user/hive/warehouse/meat/meatyear=2022  
-rw-r--r--  3 root supergroup 889305 2023-04-03 12:43 /user/hive/warehouse/meat/meatyear=2022/2022.csv
```

# ◆ HIVE – CODE 작성 및 전처리

## # 시나리오에 따른 SQL CODE 작성

1. 연도별 가구원수에 따른 평균 식대, 육류구입비, 육류가공품구입비를 구해라.

```
select year, cnt, round(avg(meals)), round(avg(meat)), round(avg(MeatProducts))
from meat
group by year, cnt
order by year, cnt;
```

2. 연도별 가구원수별 취업인원수별 식대, 육류구입비, 육류가공품구입비를 구해라.

```
select year, cnt, hiredcnt, round(avg(meals)), round(avg(meat)), round(avg(MeatProducts))
from meat
group by year, cnt, hiredcnt
order by year, cnt, hiredcnt;
```

3. 연도별 가구원수별 맞벌이가구별 식대, 육류구입비, 육류가공품구입비를 구해라.

```
select year, cnt, dualincomecnt, round(avg(meals)), round(avg(meat)), round(avg(MeatProducts))
from meat
group by year, cnt, dualincomecnt
order by year, cnt, dualincomecnt;
```

4. 연도별 학력별 식대, 육류구입비, 육류가공품구입비를 구해라.

```
select year, edocode, round(avg(meals)), round(avg(meat)), round(avg(MeatProducts))
from meat
group by year, edocode
order by year, edocode
```

```
hive> select year, cnt, round(avg(meals)), round(avg(meat)), round(avg(MeatProducts))
> from meat
> group by year, cnt
> order by year, cnt;
```

```
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_202303092038_0147, Tracking URL = http://namenode: 50030/jobdetails.jsp?jobid=job_202303092038_0147
Kill Command = /usr/local/hadoop-1.0.4/libexec/../bin/hadoop job -Dmapred.job.tracker=namenode: 9001 -kill job_202303092038_0147
Hadoop job information for Stage- 1: number of mappers: 1; number of reducers: 1
2023-04-04 09:26:12,549 Stage-1 map = 0% reduce = 0%
```

**SQL CODE**

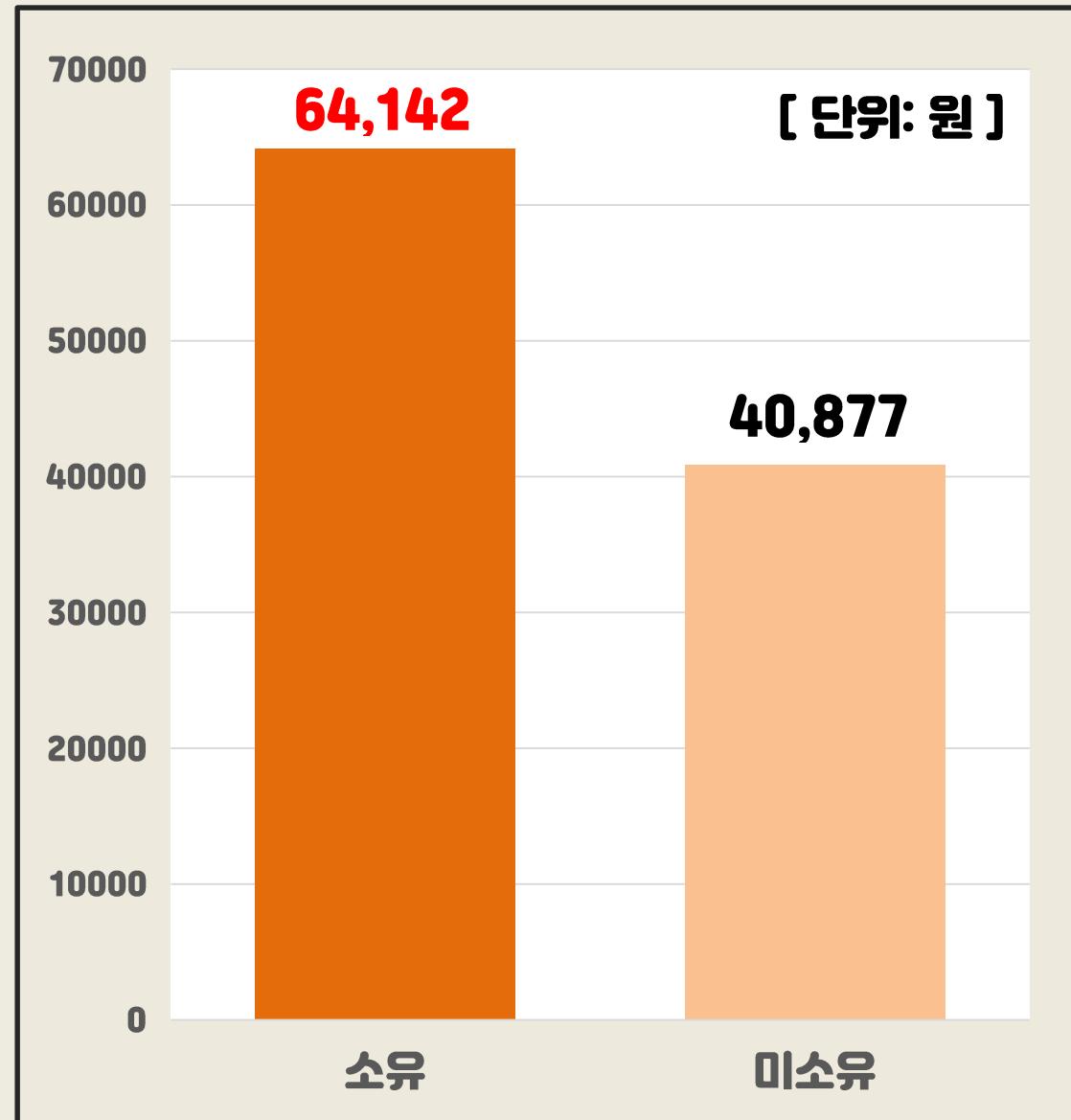
```
MapReduce Total cumulative CPU time: 5 seconds 850 msec
Ended Job = job_202303092038_0148
MapReduce Jobs Launched:
Job 0: Map: 1 Reduce: 1 Cumulative CPU: 7.41 sec HDFS Read: 6832158 HDFS Write: 2482 SUCCESS
Job 1: Map: 1 Reduce: 1 Cumulative CPU: 5.85 sec HDFS Read: 2934 HDFS Write: 1942 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 260 msec
OK
```

NULL	NULL	NULL	NULL	NULL
2014	1	188050	33362	3384
2014	2	336687	59208	7610
2014	3	666959	92863	15810
2014	4	809939	112908	21091
2014	5	852521	119802	24320
2014	6	534338	90722	19794
2014	7	488654	92505	22031
2014	8	522412	75206	9490
2014	9	503234	202417	10580
2015	1	199394	35098	3518
2015	2	318730	60527	8080
2015	3	628097	93120	15900
2015	4	845278	117743	23116
2015	5	935529	127027	26136
2015	6	500521	79769	18818
2015	7	382843	119004	19166

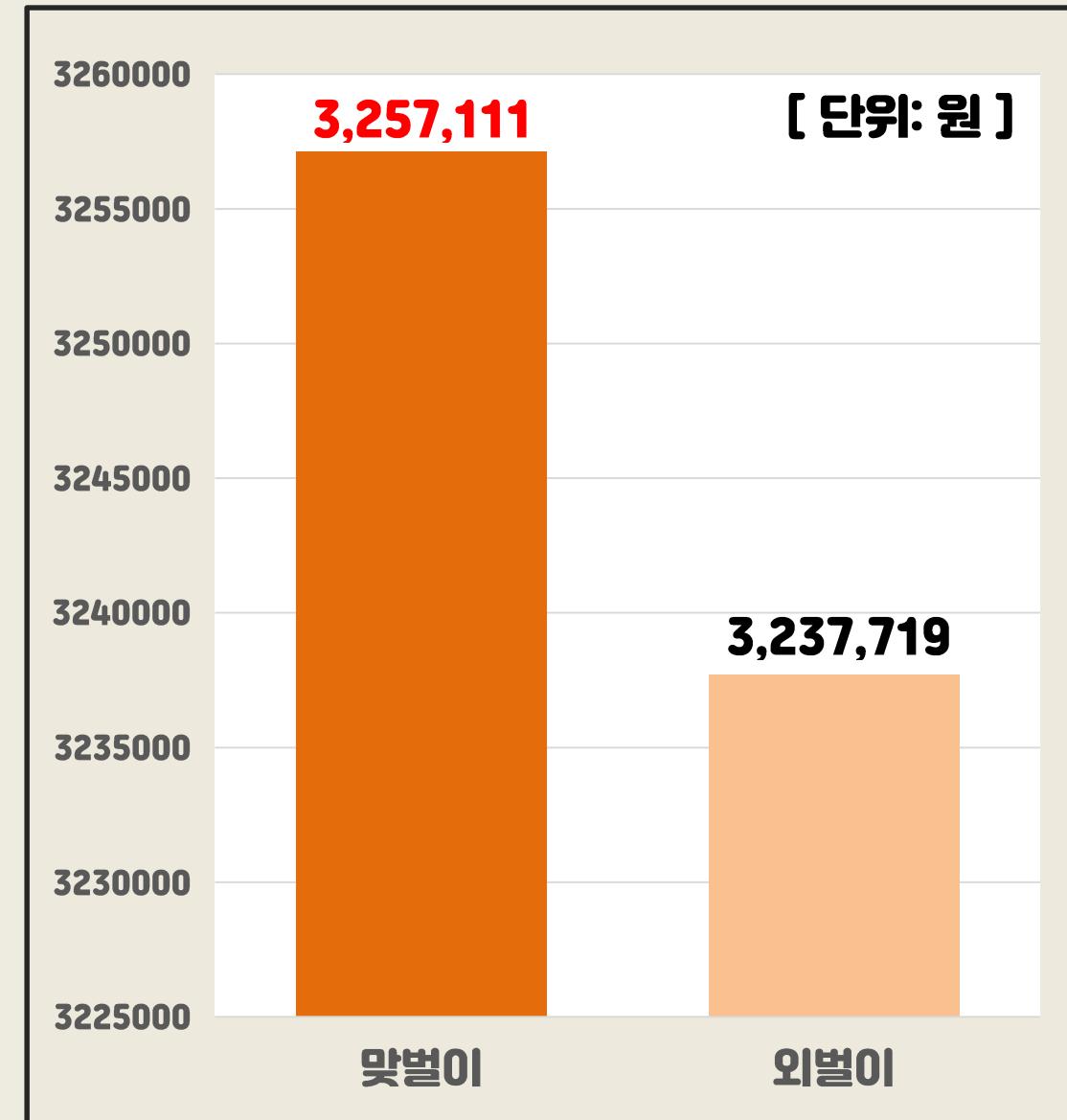
**Result [결과물]**

# 육류 구입비 in 식대

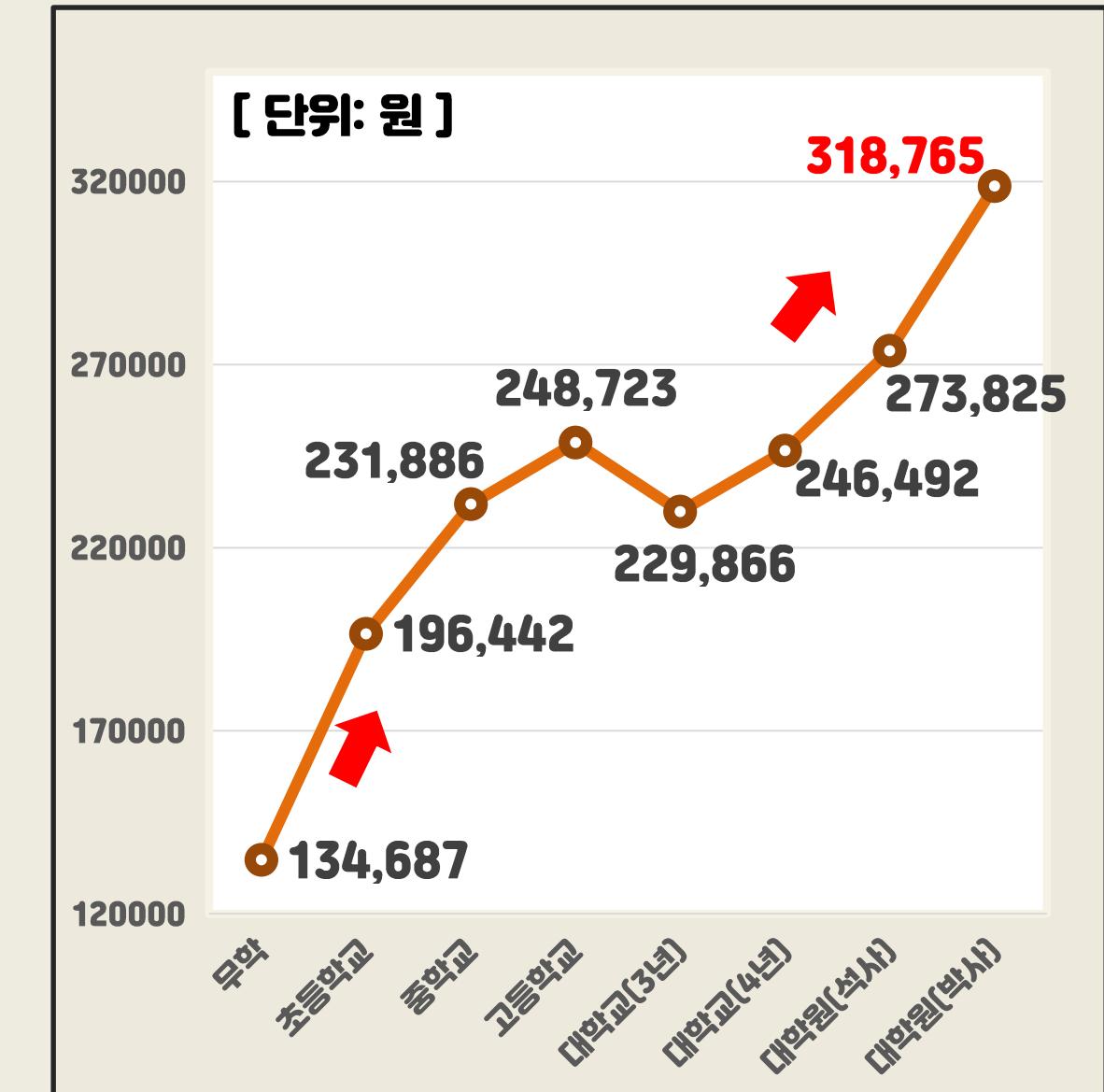
## 주택소유 / 미소유



## 맞벌이 / 외벌이



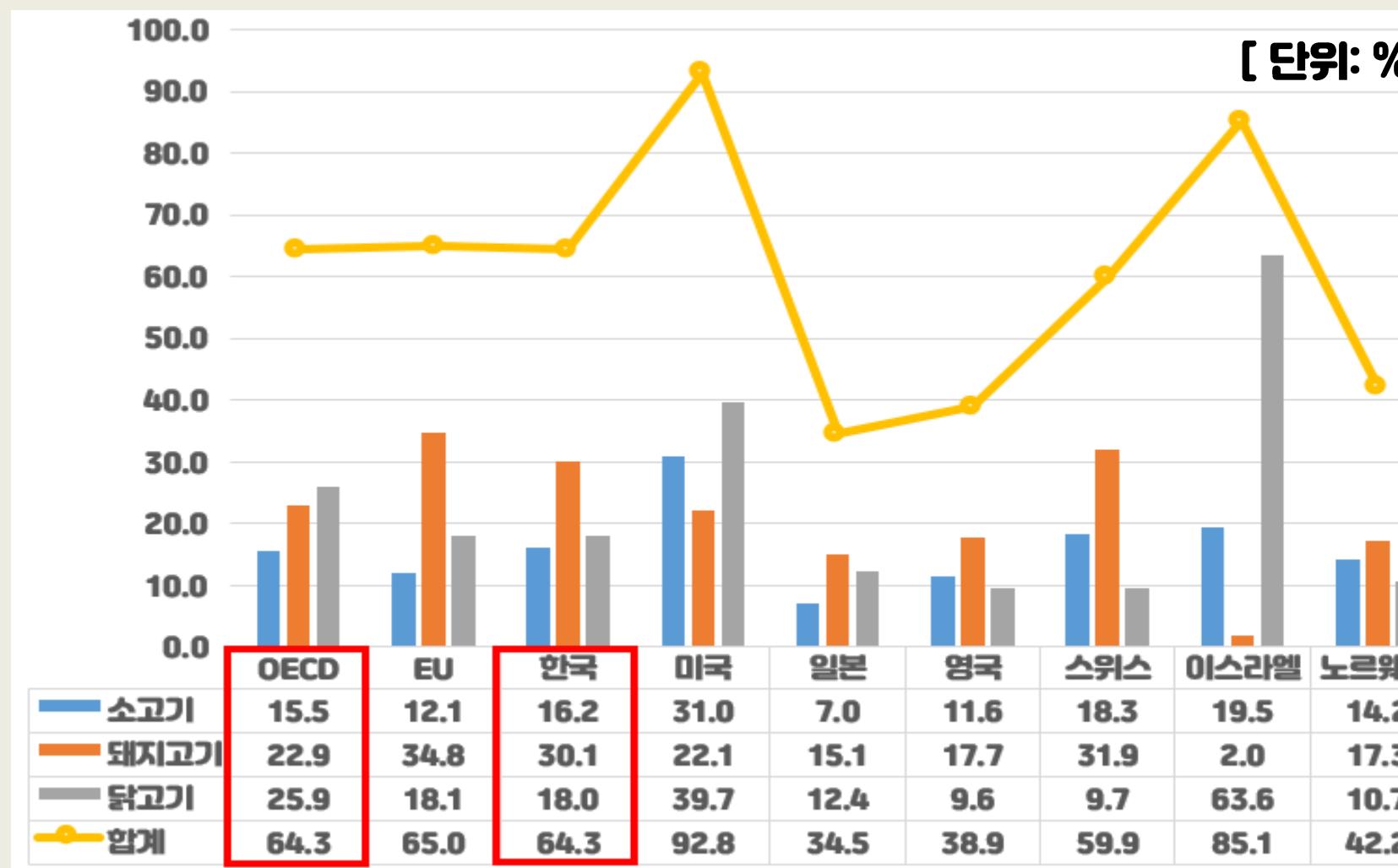
## 학력



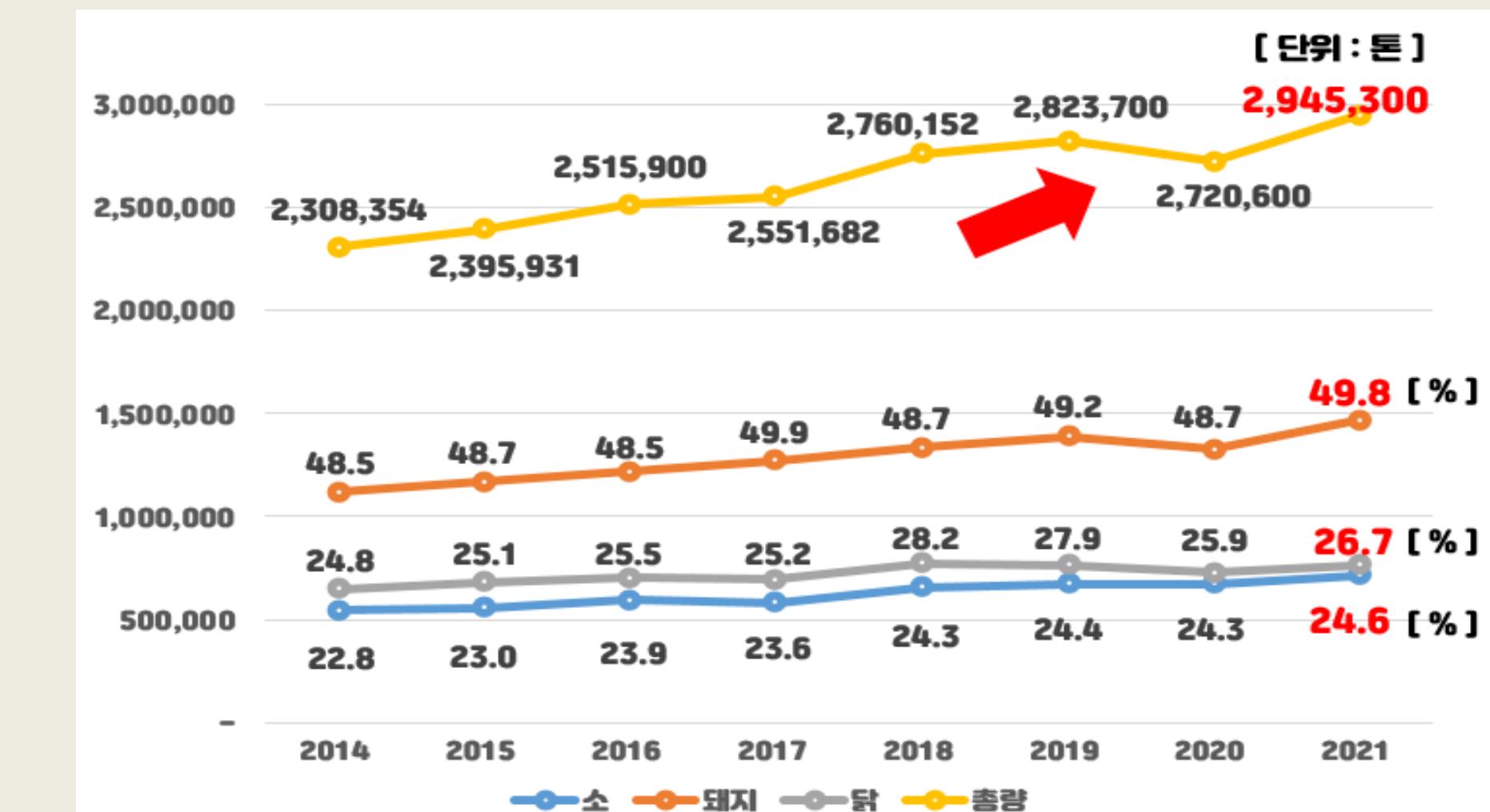
육류 구입비는 주택 소유자 / 맞벌이 / 높은 학력 일수록 높음

# 喇叭 icon 육류 소비량 분석

## OECD 평균 육류 소비량



## 국내산 육류별 소비량



평균 육류 소비량 : OECD ≈ 한국

▷ 우리나라 육류 소비량이 증가

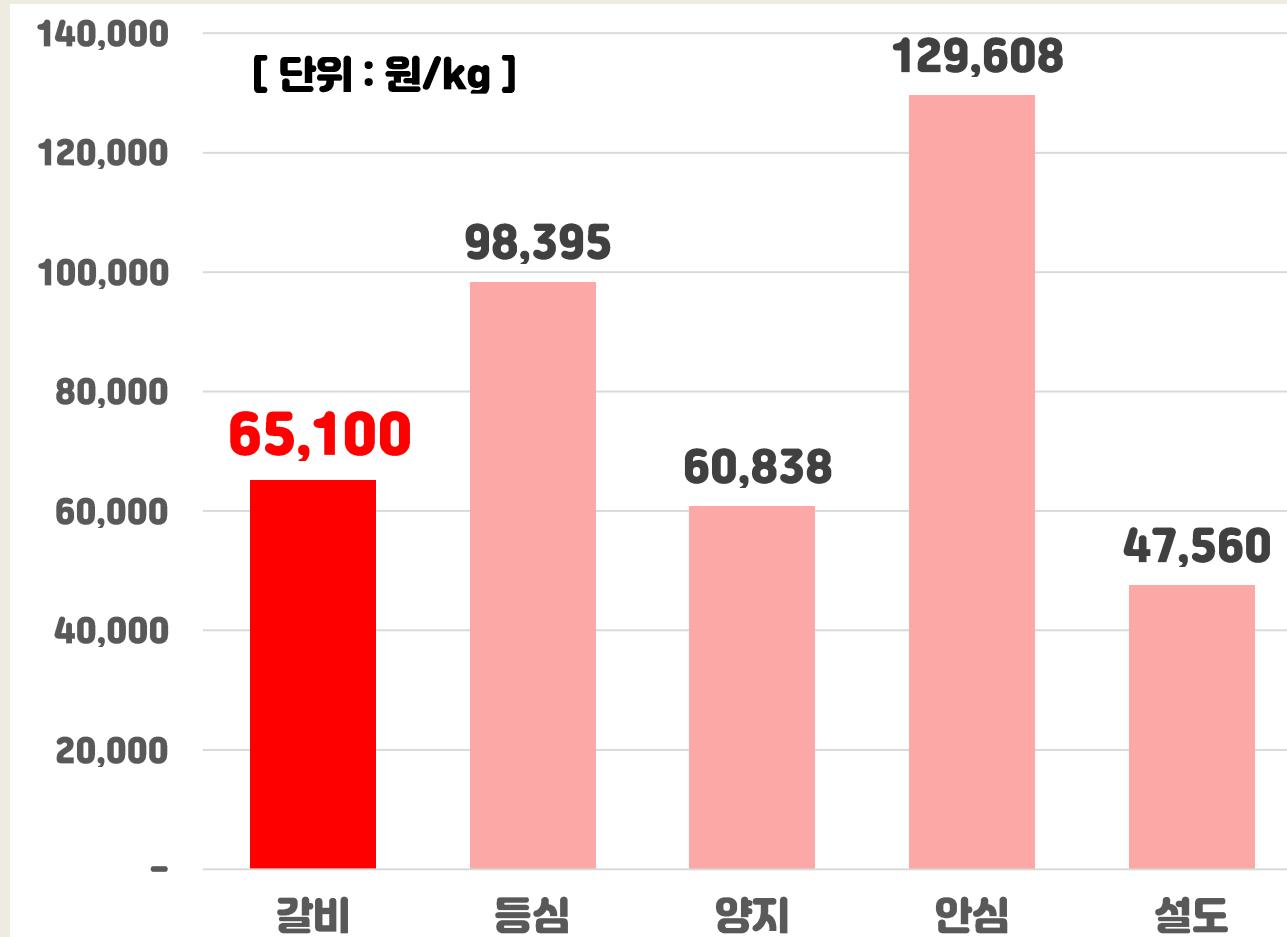
국내산 육류별 소비량 : 돼지 > 닭 > 소

돼지고기(삼겹살) 선호도 뚜렷 ▷ 돼지 소비량 ↑

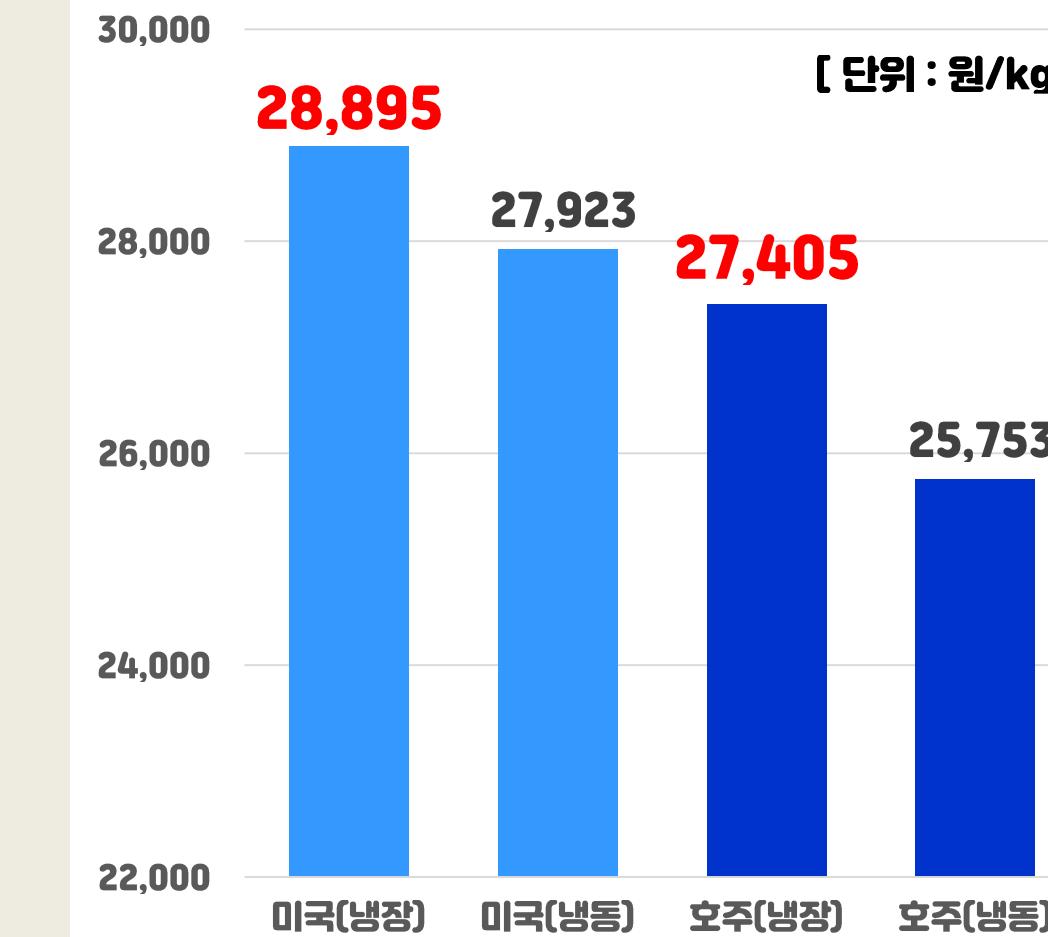


# 소고기 (Beef)

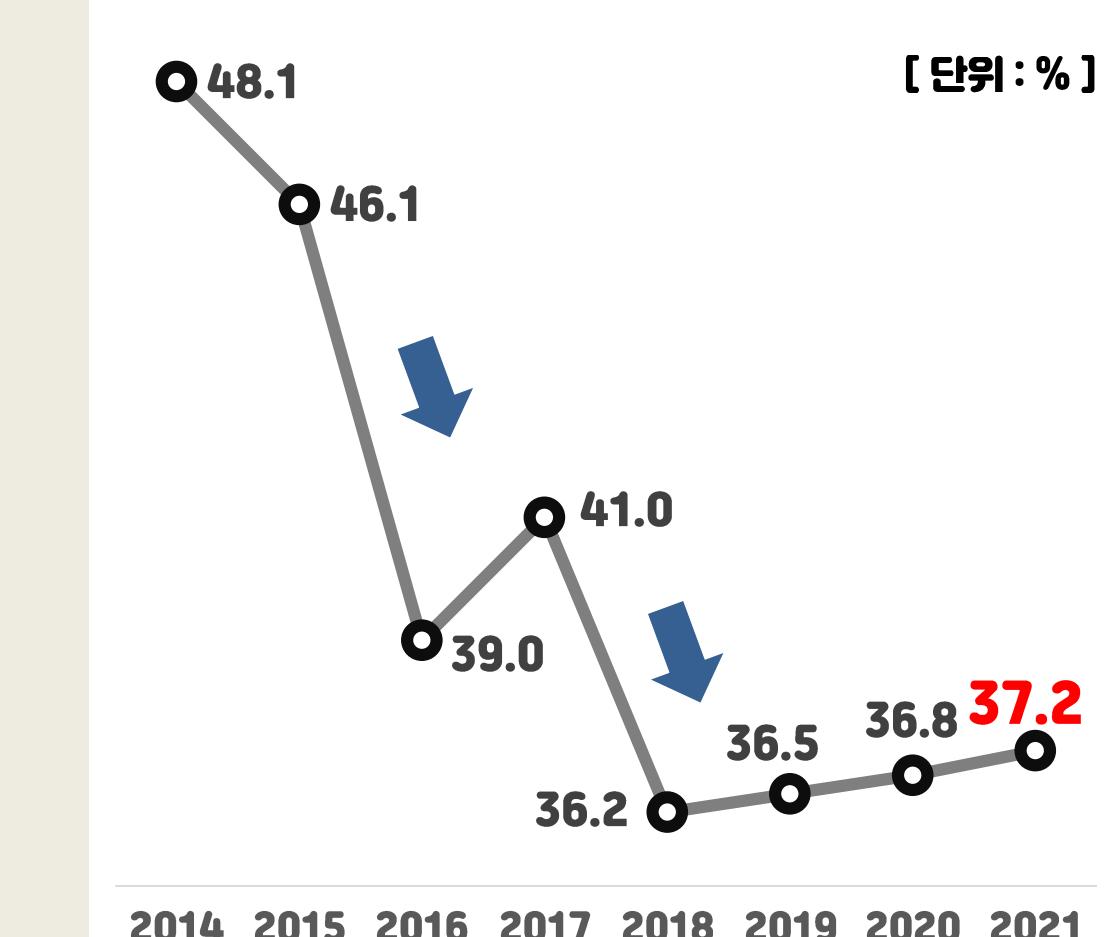
## 국내산 부위별 평균가격



## 수입산 갈비 평균가격



## 소고기 자급률



안심 > 등심 > 갈비 > 양지 > 설도 순

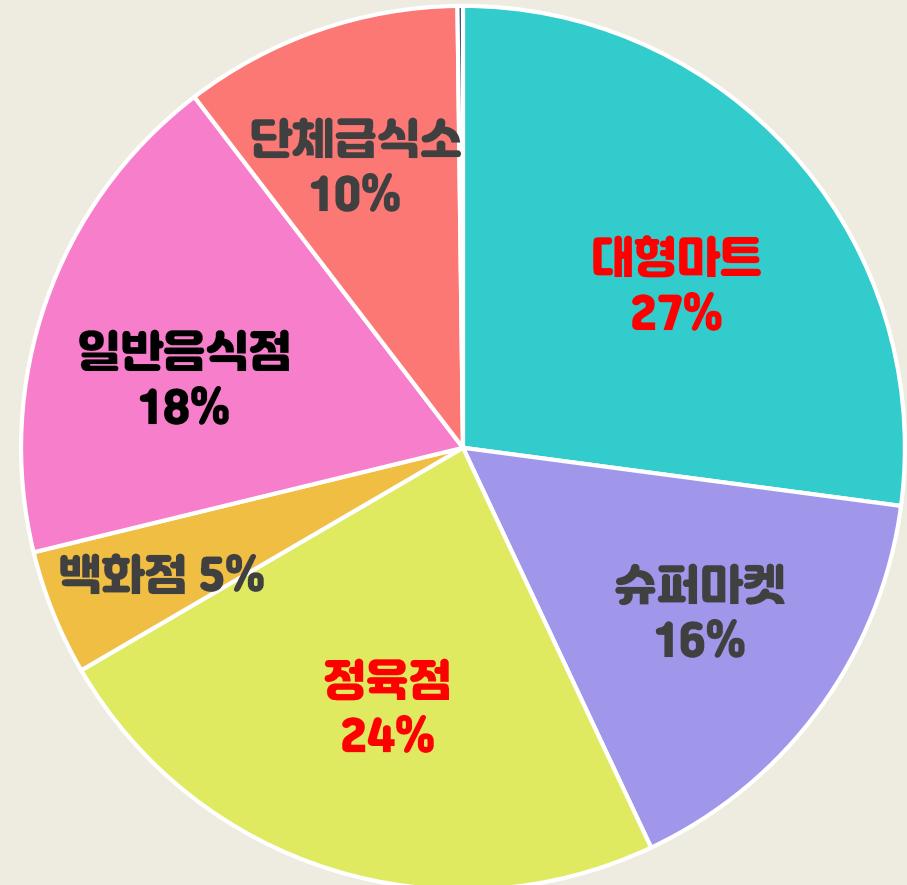
국내산 > 수입산 [약 2배]

**소고기 자급률 감소**  
\*\* 자급률 : 식품산업에서 자국산의  
비율을 가늠하는 지표



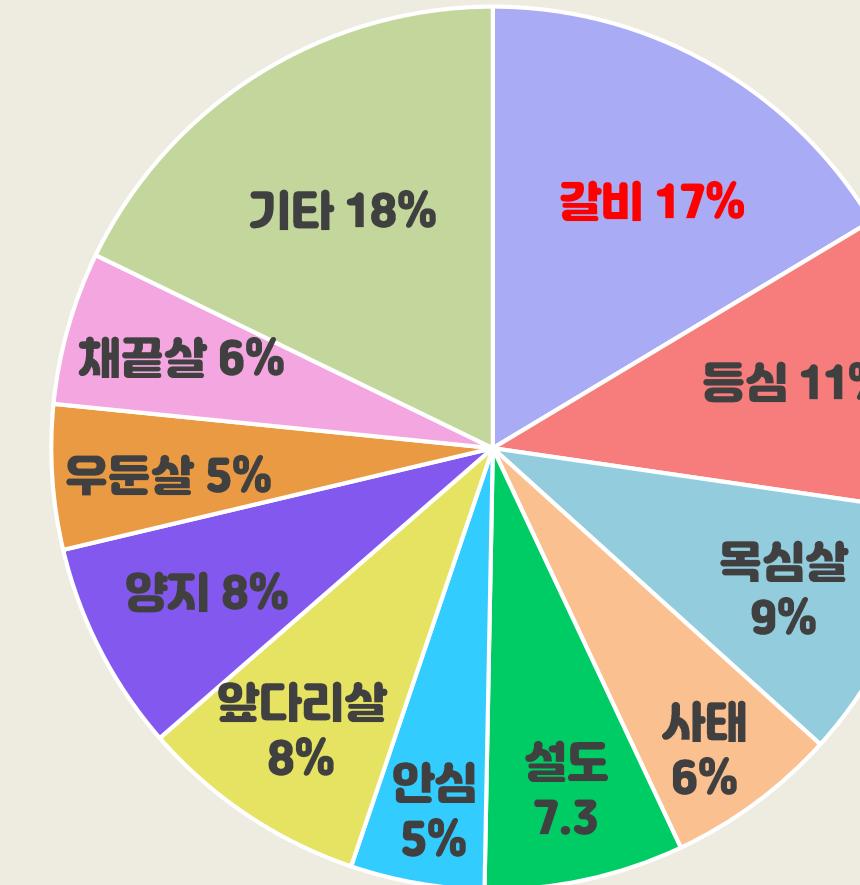
# 소고기 (Beef)

## 소매 단계별 경로 비중



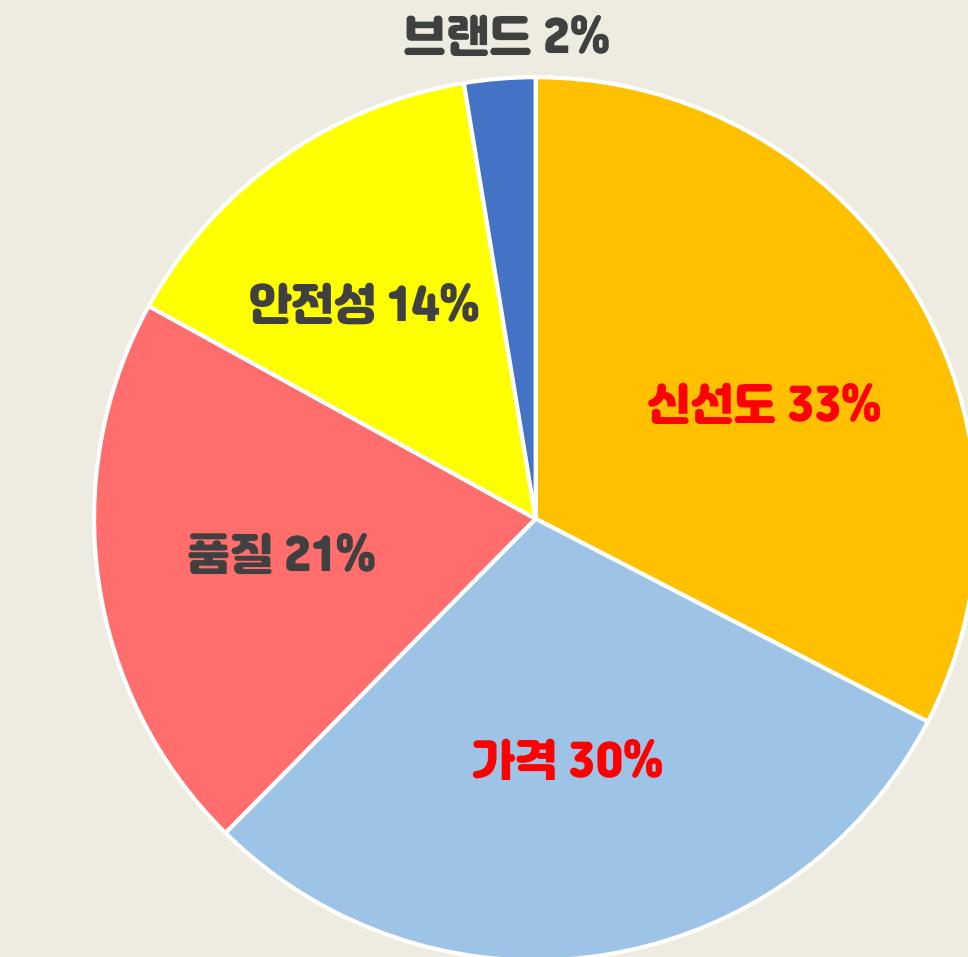
**대형마트, 정육점 주로 이용**

## 부위별 구매 비중



**부위 중 갈비 를 가장 선호**

## 구매 시 고려사항



**신선도 / 가격 을 주로 고려**



## 소고기 (Beef)

### 수입 소고기 도매가격 전망

년도	2021	2022	2023	2026	2031
환율 [원/US\$]	1,140	1,169	1,162	1,160	1,147
수입단가 [\$/kg]	미국	8.0	8.4	8.6	9.1
	호주	6.9	7.2	7.3	7.6
관세율	미국	13.3	10.7	8.0	0.0
	호주	18.7	16.00	13.3	5.3
도매원가 [원/kg]	미국	11,314	11,828	11,734	11,584
	호주	10,105	10,574	10,499	10,398
					9,607

#### - 국내산 공급량 증가 추세

But, 관세율 철폐로 인한 수입산 소고기 가격 경쟁력 강화

▶ 관세 철폐: 미국 2026년 / 호주 2031년

▶ 소고기 자급률 하락으로 연결

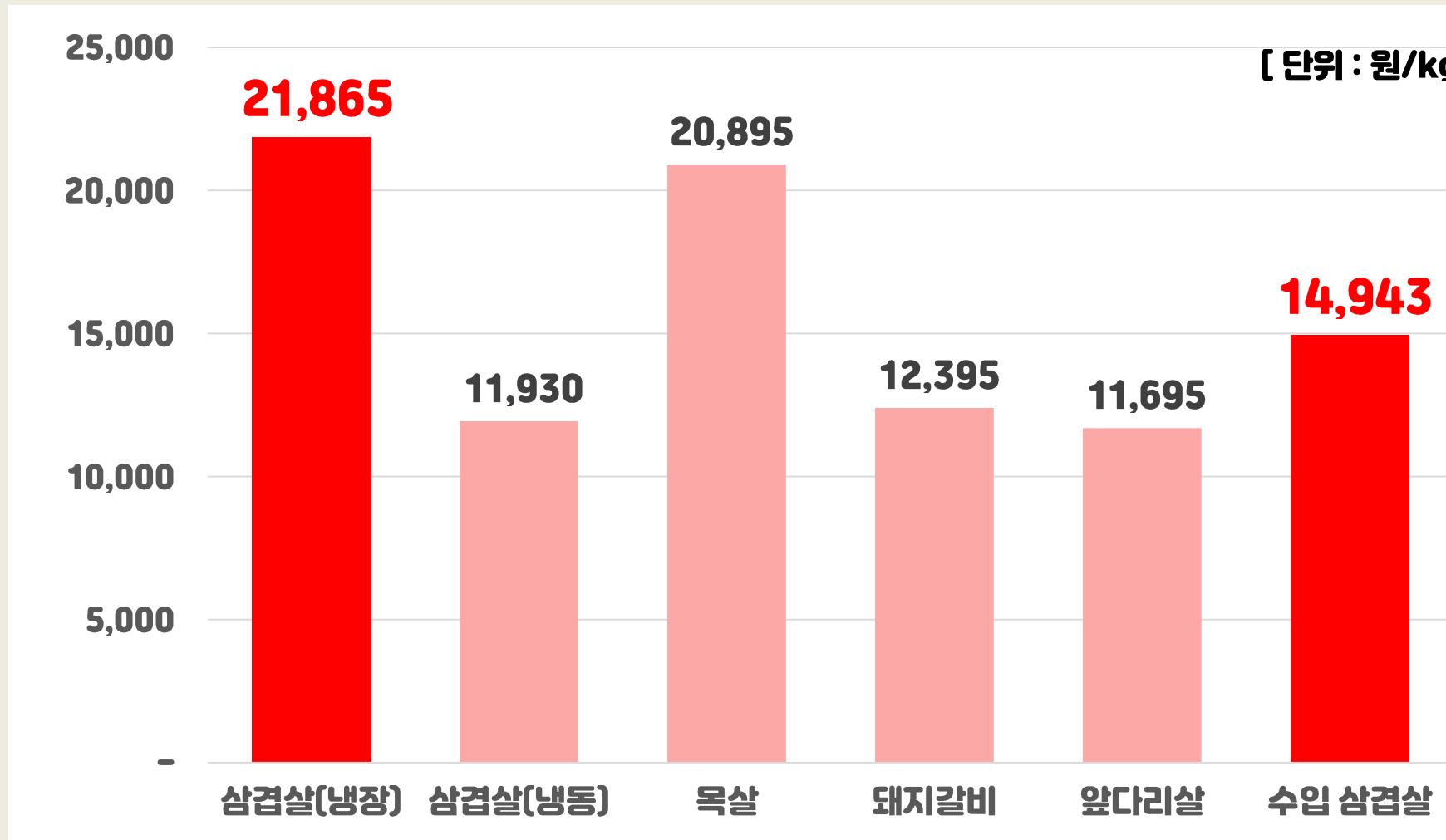
▶ 국내산 가격 경쟁력 강화 대책 마련 필요

=> 선제적이고 자율적인 수급 조절

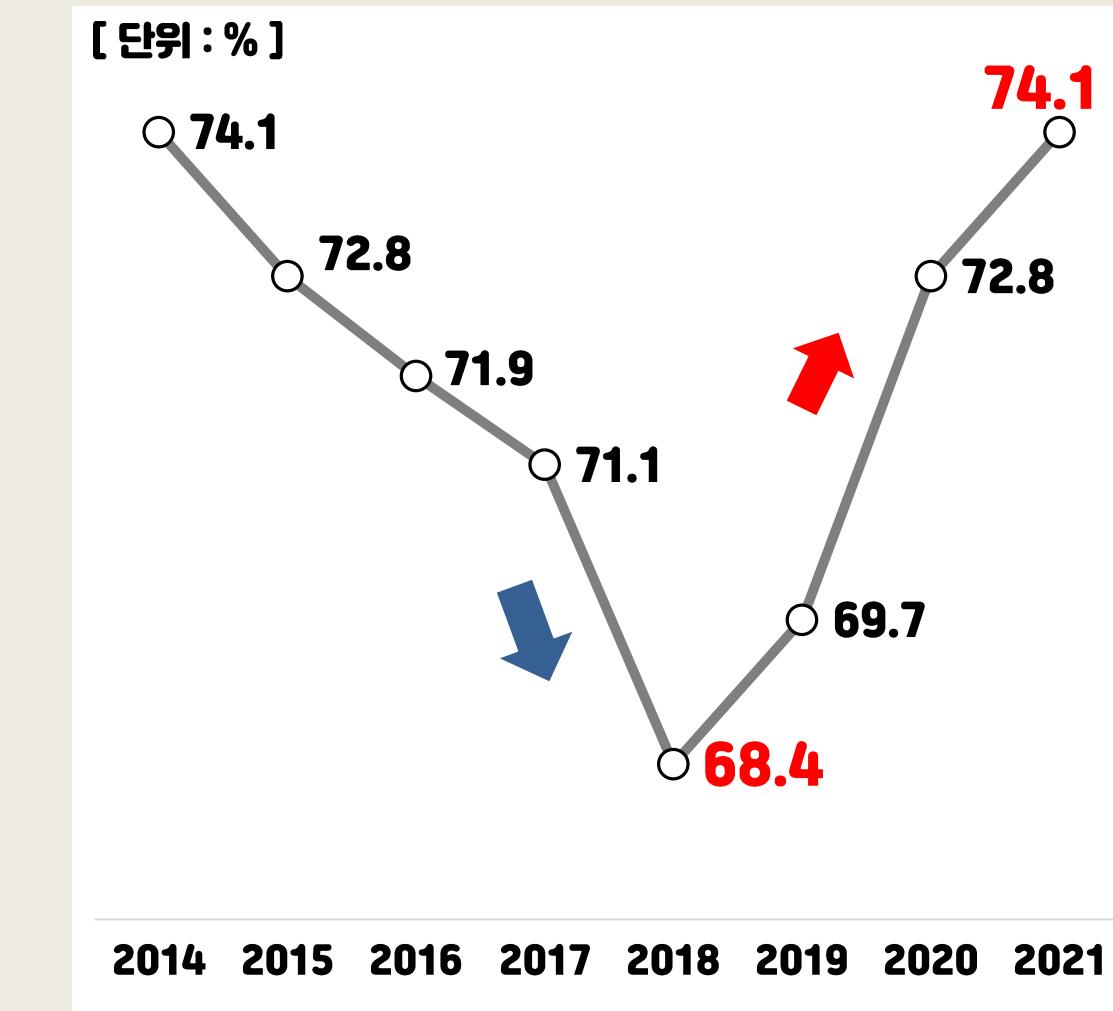


## 돼지고기 (Folk)

### 국내산, 수입산 부위별 평균가격



### 자급률



삼겹살(냉장) > 목살 > 수입 삼겹살 > 삼겹살(냉동) 순

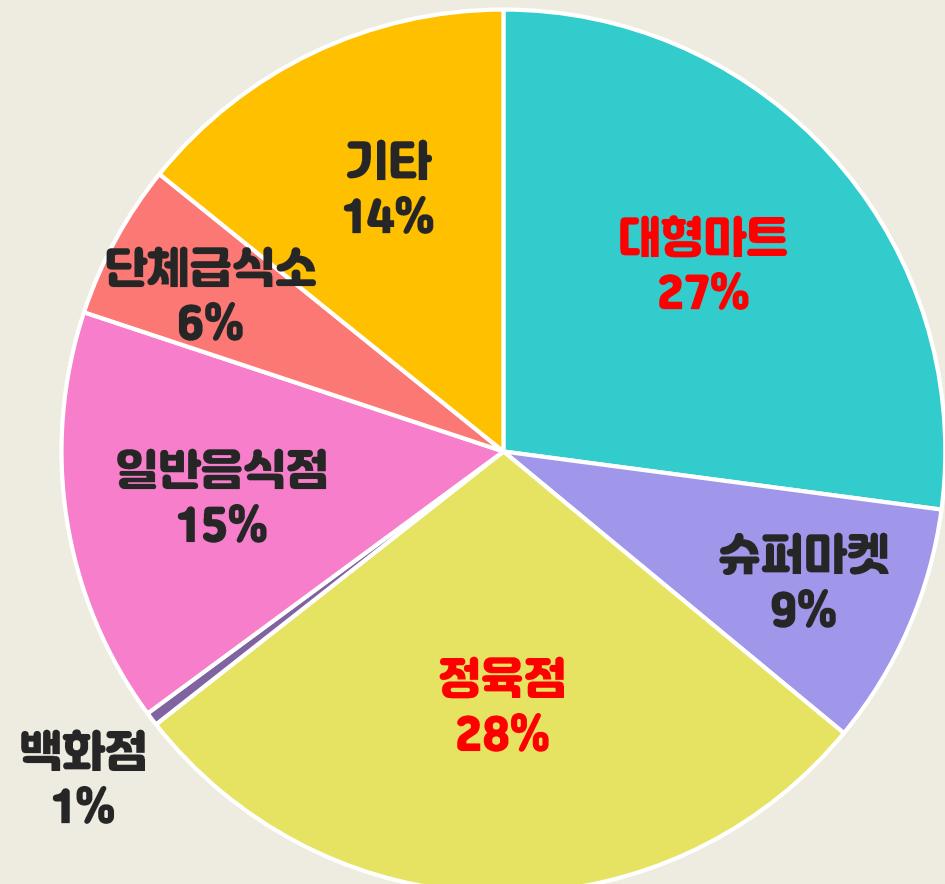
돼지고기 자급률 감소 후 증가 추세

\* 2018년 : ASF (아프리카 돼지열병)



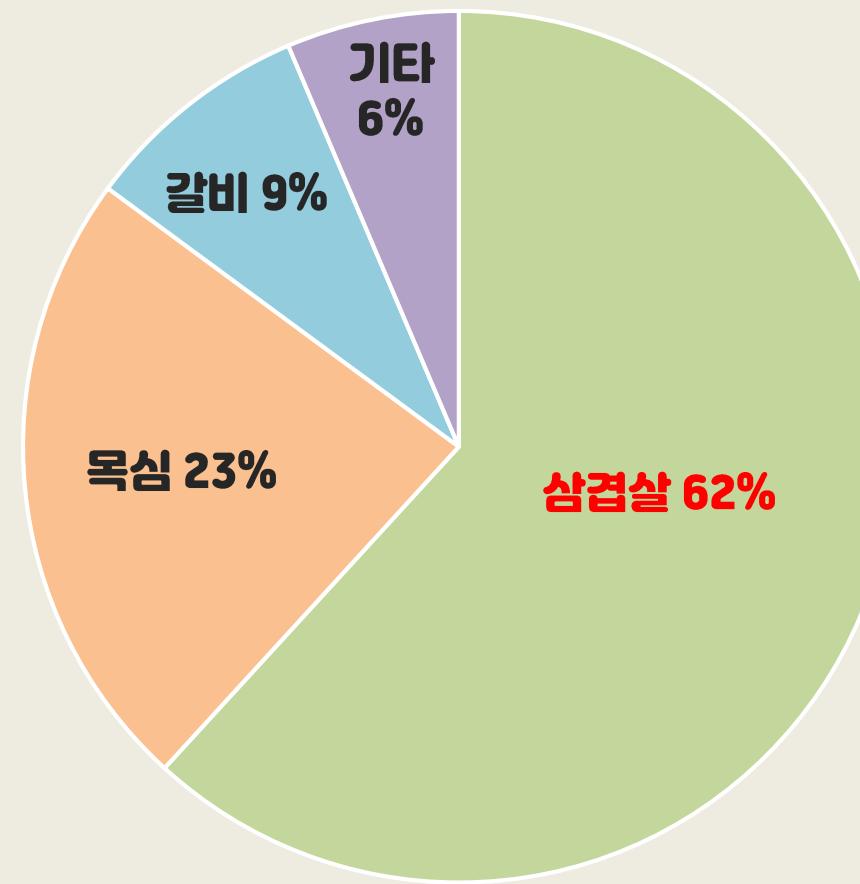
## 돼지고기 (Folk)

**소매 단계별 경로 비중**



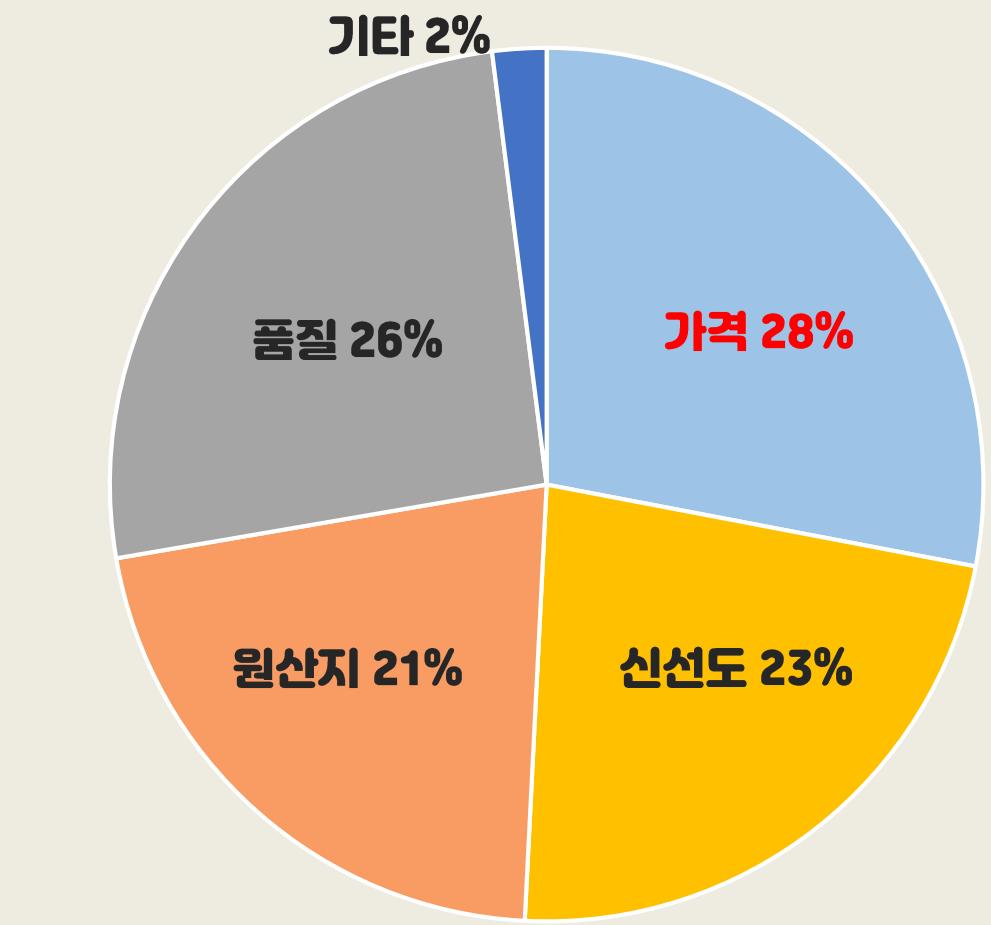
**대형마트, 정육점 주로 이용**

**부위별 구매 비중**



**부위 중 삼겹살을 가장 선호**

**구매 시 고려사항**



**가격을 주로 고려  
품질 / 신선도 / 원산지도 중요**



## 돼지고기 (Folk)

### 수입 돼지고기 도매가격 전망

년도	2021	2022	2023	2026	2031
환율 [원/US\$]	1140.0	1169.0	1162.0	1160.0	1147.0
수입단가 [\$/kg]	미국	3.2	2.8	2.7	2.8
	EU	4.7	4.1	3.9	4.0
	기타	3.1	2.8	2.7	2.7
관세율	미국	0.0	0.0	0.0	0.0
	EU	0.0	0.0	0.0	0.0
	기타	6.2	6.2	6.2	6.2
도매원가 [원/kg]	미국	4,051	3,642	3,492	3,534
	EU	5,842	5,253	5,037	5,097
	기타	4,167	3,747	3,593	3,636
					3,584

- 국내산 공급량 증가 추세

But, 관세율 철폐로 인한 수입산 돼지고기 가격 경쟁력 강화

▶ 관세 철폐: 미국, EU 2021년 이미 시행

▶ ASF / 수입량 증감에 따라 자급률 변동 가능성 있음

▶ 국내 돼지고기 국제 경쟁력 낮은 편

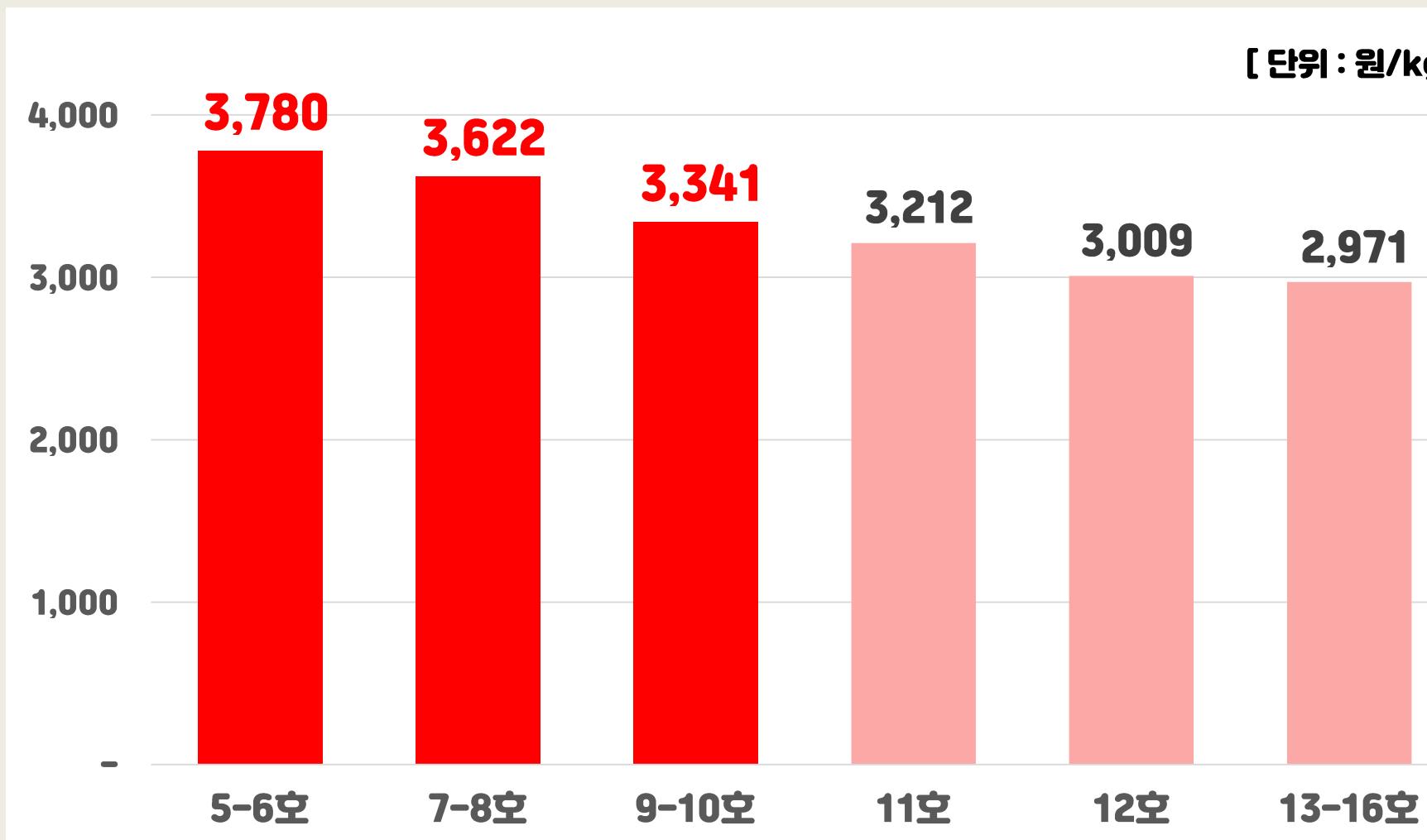
=> 경쟁력 강화 대책 강구 필요

(지속적인 입식 조절 및 생산성 향상 노력)



# 닭고기 (Chicken)

## 호수별 평균가격

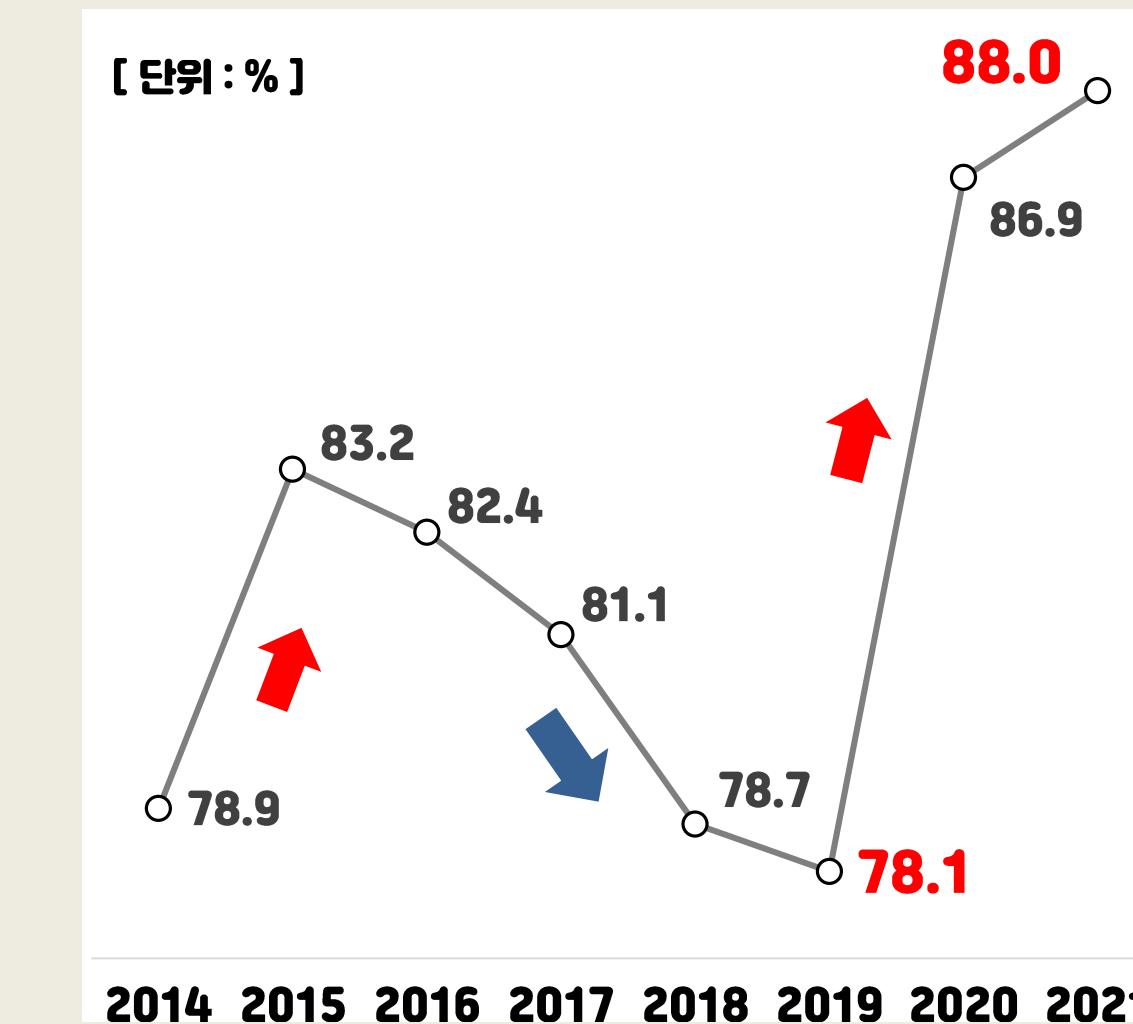


### 닭고기 관련 사업장 수요 영향

5-6호 : 삼계탕 용

7-10호 : 프랜차이즈 치킨 용

## 자급률



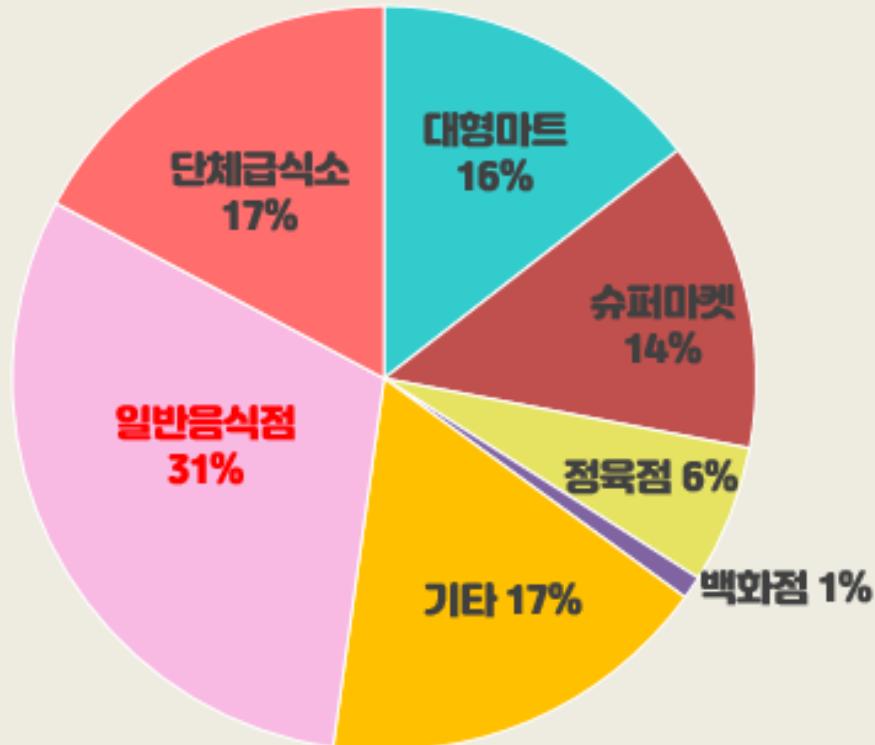
### 닭고기 자급률 감소 후 증가 추세

\* 2019년 : HPAI (고병원성 조류인플루엔자)

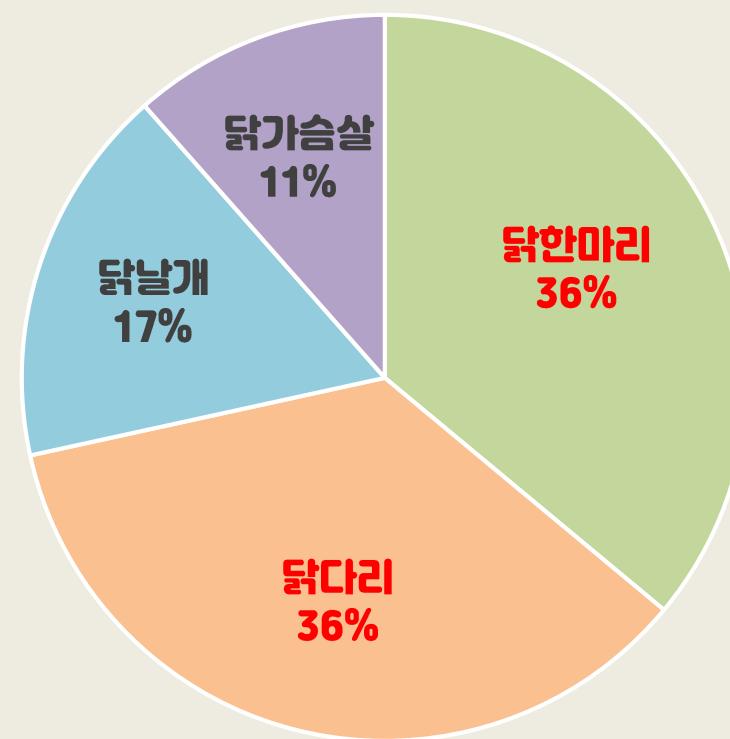


# 닭고기 (Chicken)

## 소매 단계별 경로 비중



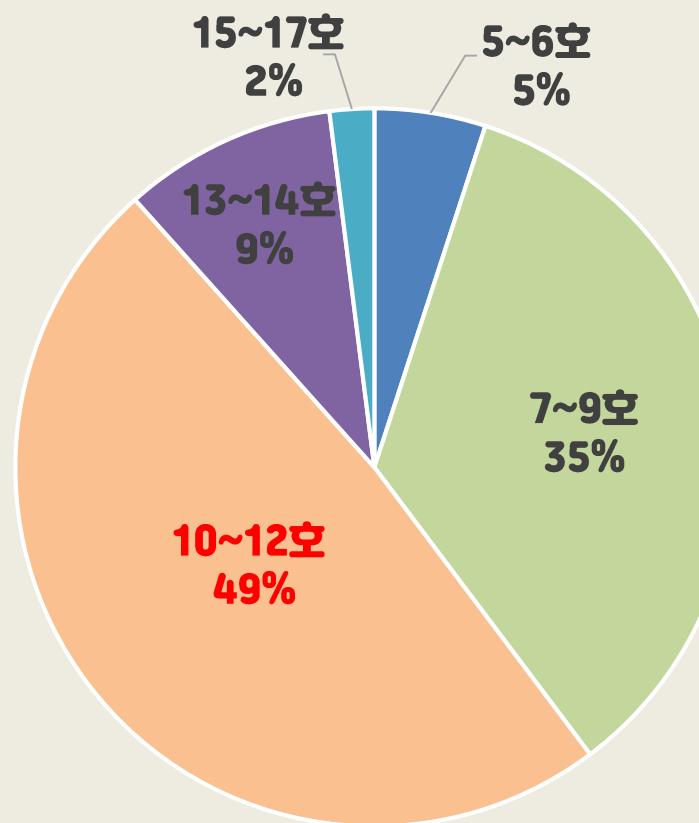
## 부위별 구매 비중



**일반음식점** 을 주로 이용

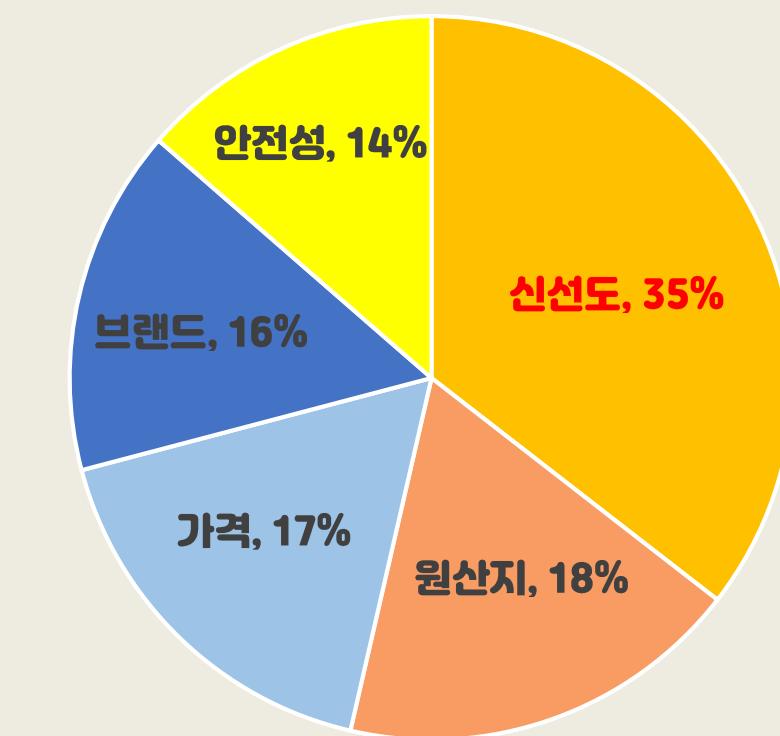
**닭한마리, 닭다리** 위주  
구매 선호

## 생닭 구매 시 선호 크기



**10~12호** 구매 선호

## 구매 시 고려사항



**신선도** 를 주로 고려



# 닭고기 (Chicken)

## 수입 닭고기 도매가격 전망

년도	2021	2022	2026	2031
환율 (원/US\$)	1,140	1,169	1,160	1,147
수입단가 (\$/kg)	브라질	1.69	1.64	1.70
	미국	2.62	2.55	2.63
	덴마크	1.80	1.75	1.81
관세율	브라질	20.0	20.0	20.0
	미국	0.0	0.0	0.0
	덴마크	4.3	2.9	0.0
도매원가 (원/kg)	브라질	2,503	2,496	2,559
	미국	3,287	3,279	3,361
	덴마크	2,348	2,313	2,310
				2,492
				3,273
				2,250

- 국내산 공급량 증가 추세

But, 관세율 철폐로 인한 수입산 닭고기 가격 경쟁력 강화

▶ 관세 철폐 : 미국 2021년 이미 시행 / 덴마크 2026년

▶ HPAI / 병아리 수급 불균형에 따라 자급률 변동 가능성 있음

▶ 생산비 상승 및 생계유통가격 하락으로 인한 계업업체 부담

=> 경쟁력 강화 대책 강구 필요 (종계 및 육계 생산성 향상 등)

# 1. 육류 소비 구조 변화에 따른 대응

## 가정 내 소비 / 1·2인 가구 증가



**소포장 판매 경로 확대**  
(정육 자판기, 편의점)

## 고령층 인구 증가



**고령 친화 식품 개발**

## 소고기 효율적인 유통체계



**도소매 가격 연동제 필요**  
\*\* 23년 기준 : 도매가격 23.6% ↓ / 소매가격 8.7% ↓

## 2. 국내 육류 경쟁력 확보

### 스마트 축사( ICT 융합 )



**온·습도 조절, 사료 공급 자동화, 악취제거**  
 ▶ **생산비 절감 / 가격 경쟁력 강화 / 전염병, 폐사율 감소 유도**

### 육류 원산지 검정 키트



**육류별 원산지 검정 키트 적용**  
 ▶ **국내산 육류 안전성 확보**  
 \*\*돼지 외 다른 육류에 확대 적용 필요

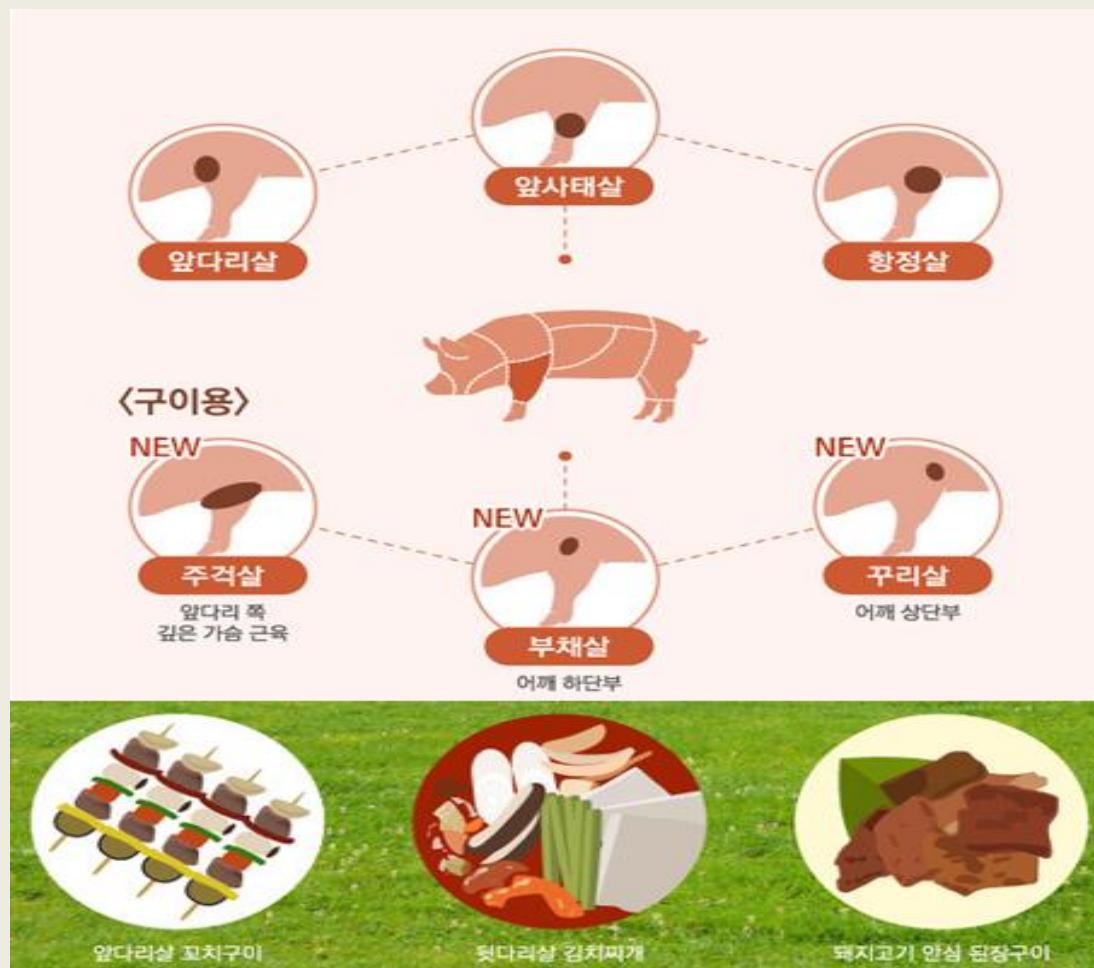
### 3. 비선호 부위 재고 문제 해결

#### 비선호 부위 활용한 조리법 개발



**이탈리안 소시지 ( 살라미 )  
라르도 ( Lardo )**

#### 저지방 부위 활용법 제시



**저지방 / 맛↑ / 가격↓**

#### 육류가공 제품 개발 by PEF



**PEF (고전압펄스전기장) 이용한  
고기 육질 연화  
\*\*기술 상용화 필요**

## 4. 대체육 개발로 육류 수요량 충족

### 가치소비 인식 확산



**간헐적 채식주의자  
(플렉시테리언) 증가 추세**

### 소비자 맞춤 판매 전략



- 냉장 제품 선호 (냉동 X)
- 대체육 선호 그룹 타겟 마케팅

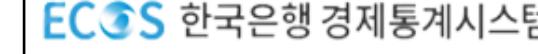
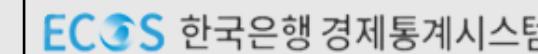
### 대체육 비선호자 고려



- 대체육 편견 요소 제거**
- 실제 육류와 유사한 맛 / 식감
- 제품 개발 및 홍보



# 데이터 목록 및 출처

Index	사용 데이터	파일형식	출처
1	가계동향조사	csv	 MicroData Integrated Service
2	장래인구추계	csv	 통계청 국가통계포털
3	식대 지출액 중 육류 / 육류 가공품 비중	csv	 통계청 국가통계포털
4	1인당 국민소득(GNI)	csv	 한국은행 경제통계시스템
5	국내총생산(GDP)	csv	 한국은행 경제통계시스템
6	농림축산식품 주요 통계	pdf	 농림축산식품부
7	축산물 가격 및 수급 자료	pdf	 농협경제지주
8	축산물 유통 실태	pdf	 축산물품질평가원 Korea Institute for Animal Products Quality Evaluation
9	OECD 국가 육류 소비량 (meat-consumption)	pdf	 OECD



## 참고문헌

[1] 정민국 · 김현중 · 이형우, 2020 「육류 소비행태 변화와 대응과제」

[2] 이형우 · 지선우, 2022 「한육우, 돼지, 젖소 수급 동향과 전망」

[3] 이형우 · 김서영, 2022 「가금류 수급 동향 및 전망」

[4] 이형우 · 김형진, 2020 「한육우, 돼지, 젖소 수급 동향과 전망」

[5] 이형우 · 심민희, 2020 「가금류 수급 동향과 전망」

[6] 김계웅 · 김석은, 2009 「국내 소비자의 돼지고기 선호도와 소비행태 분석」

[7] 퇴근시간 / 법인카드 이용시간 비중 변화, 2020 : <https://www.shinhancardblog.com/>

[8] 식물성 대체육 시장 : <https://www.donga.com/news/Economy/article/all/20220801/114747789/1>

[9] 돼지고기 원산지 검정키트 : <https://www.fnnews.com/news/201410221507194503>

[10] 도 · 소매가격 연동제 : <https://www.agrinet.co.kr/news/articleView.html?idxno=316027>



## 분석도구





# 감사합니다

