빅데이터최적화개론

20231275 신민서

jupyter lotto.csv 11분 전

File Edit View Language

```
Round,Num1,Num2,Num3,Num4,Num5,Num6,Bonus,First-price,First-winners,Second-price,Second-winners
1146,6,11,17,19,40,43,28,11,"2,526,476,353",72,"64,331,574"
1145,2,11,31,33,37,44,32,9,"3,051,630,084",63,"72,657,860"
1144,3,4,12,15,26,34,6,18,"1,489,347,375",94,"47,532,634"
1143,10,16,17,27,28,36,6,"2,545,657,023",11,"51,855,977",90
1142,2,8,28,30,37,41,22,"3,117,517,709",9,"69,795,173",67
1141,7,11,12,21,26,35,20,"2,457,758,285",11,"45,058,902",100
1140,7,10,22,29,31,38,15,"2,279,823,938",12,"54,935,517",83
1139,5,12,15,30,37,40,18,"2,167,490,972",13,"62,616,406",75
1138,14,16,19,20,29,34,35,"1,902,656,786",14,"88,790,650",50
1137,4,9,12,15,33,45,26,"2,023,447,688",14,"42,534,937",111
1136,21,33,35,38,42,44,1,"2,314,468,157",12,"61,719,151",75
1135,1,6,13,19,21,33,4,"2,953,726,125",9,"48,687,794",91
1134,3,7,9,13,19,24,23,"1,755,689,384",14,"42,233,078",97
1133,13,14,20,28,29,34,23,"2,105,079,491",13,"62,479,529",73
1132,6,7,19,28,34,41,5,"2,404,951,807",11,"60,398,334",73
1131,1,2,6,14,27,38,33,"1,542,367,898",17,"78,036,471",56
1130,15,19,21,25,27,28,40,"2,263,301,219",12,"65,602,934",69
1129,5,10,11,17,28,34,22,"2,369,567,660",11,"40,600,069",107
1128,1,5,8,16,28,33,45,"419,925,560",63,"57,262,577",77
1127,10,15,24,30,31,37,32,"2,267,891,969",12,"53,997,428",84
1126,4,5,9,11,37,40,7,"2,386,382,421",11,"48,077,302",91
1125,6,14,25,33,40,44,30,"2,195,289,188",12,"56,289,467",78
1124,3,8,17,30,33,34,28,"2,623,327,913",10,"50,255,324",87
1123,13,19,21,24,34,35,26,"1,731,310,711",16,"59,958,813",77
1122,3,6,21,30,34,35,22,"2,556,266,046",11,"47,338,261",99
1121,6,24,31,32,38,44,8,"2,524,513,262",11,"60,898,347",76
1120,2,19,26,31,38,41,34,"2,522,163,375",11,"50,812,816",91
1119,1,9,12,13,20,45,3,"1,396,028,764",19,"45,574,823",97
1118,11,13,14,15,16,45,3,"1,477,445,132",19,"64,980,226",72
1117,3,4,9,30,33,36,7,"3,028,385,542",9,"48,325,302",94
1116,15,16,17,25,30,31,32,"2,695,000,238",10,"39,400,589",114
```

순위	등위별 총 당첨금액	당첨게임 수	1게임당 당첨금액	당첨기준	비고
1등	27,791,239,883원	11	2,526,476,353원	당첨번호 **6개** 숫자일치	1등 자동8 수동2 반자동1
2등	4,631,873,328원	72	64,331,574원	당첨번호 **5개** 숫자일치 +**보너스** 숫자일치	
3등	4,631,876,160원	3,030	1,528,672원	당첨번호 **5개** 숫자일치	
4등	7,907,300,000원	158,146	50,000원	당첨번호 **4개** 숫자일치	
5등	13,385,085,000원	2,677,017	5,000원	당첨번호 **3개** 숫자일치	

- 당첨금 지급기한 : 지급개시일로부터 1년 (휴일인 경우 익영업일)
- 총판매금액 : 116,694,743,000원

당첨판매점 조회 | 미수령 당첨금 안내 | 1146 부터 1146 까지 | 인쇄하기 | 엑셀다운로드

csv 파일 불러오고 파일 특징 파악하기

In [2]:
```
df = pd.read_csv("lotto.csv")
```

In [3]:
```
df.isna().sum()
```

Out[3]:
```
Round            0
Num1             0
Num2             0
Num3             0
Num4             0
Num5             0
Num6             0
Bonus            0
First-price      0
First-winners    0
Second-price     0
Second-winners   0
dtype: int64
```

In [4]:
```
df.head()
```

Out[4]:

	Round	Num1	Num2	Num3	Num4	Num5	Num6	Bonus	First-price	First-winners	Second-price	Second-winners
0	1146	6	11	17	19	40	43	28	2,526,476,353	11	64,331,574	72
1	1145	2	11	31	33	37	44	32	3,051,630,084	9	72,657,860	63
2	1144	3	4	12	15	26	34	6	1,489,347,375	18	47,532,634	94
3	1143	10	16	17	27	28	36	6	2,545,657,023	11	51,855,977	90
4	1142	2	8	28	30	37	41	22	3,117,517,709	9	69,795,173	67

로또 파일에서 일반 번호와 보너스 번호 분리하기

In [5]:
```
LDN = df.loc[:,["Round","Num1","Num2", "Num3", "Num4", "Num5", "Num6"]]
LDN.index = LDN.index[::-1]
LDN = LDN.sort_index()
LDN
```

Out[5]:

	Round	Num1	Num2	Num3	Num4	Num5	Num6
0	1	10	23	29	33	37	40
1	2	9	13	21	25	32	42
2	3	11	16	19	21	27	31
3	4	14	27	30	31	40	42
4	5	16	24	29	40	41	42
...	...	...	...	...	...	...	...
1141	1142	2	8	28	30	37	41
1142	1143	10	16	17	27	28	36
1143	1144	3	4	12	15	26	34
1144	1145	2	11	31	33	37	44
1145	1146	6	11	17	19	40	43

1146 rows × 7 columns

In [6]:
```
LDB = df.loc[:,["Round", "Bonus"]]
LDB.index = LDB.index[::-1]
LDB = LDB.sort_index()
LDB
```

Out[6]:

	Round	Bonus
0	1	16
1	2	2
2	3	30
3	4	2
4	5	3
...	...	...
1141	1142	22
1142	1143	6
1143	1144	6
1144	1145	32
1145	1146	28

1146 rows × 2 columns

```
In [7]: LDN_T = LDN.loc[:,["Num1","Num2", "Num3", "Num4", "Num5", "Num6"]]
        LDB_T = LDB.loc[:,["Bonus"]]
```

```
In [8]: LDN_TC = LDN_T.apply(pd.Series.value_counts).sum(axis=1).sort_values(ascending=False)
        LDN_TC
```

```
Out[8]: 34    177.0
        12    168.0
        33    165.0
        18    165.0
        13    165.0
        14    165.0
        45    164.0
        40    163.0
        17    161.0
        27    161.0
        37    159.0
        20    159.0
        43    158.0
        19    158.0
        1     158.0
        11    157.0
        21    157.0
        3     156.0
        7     156.0
```

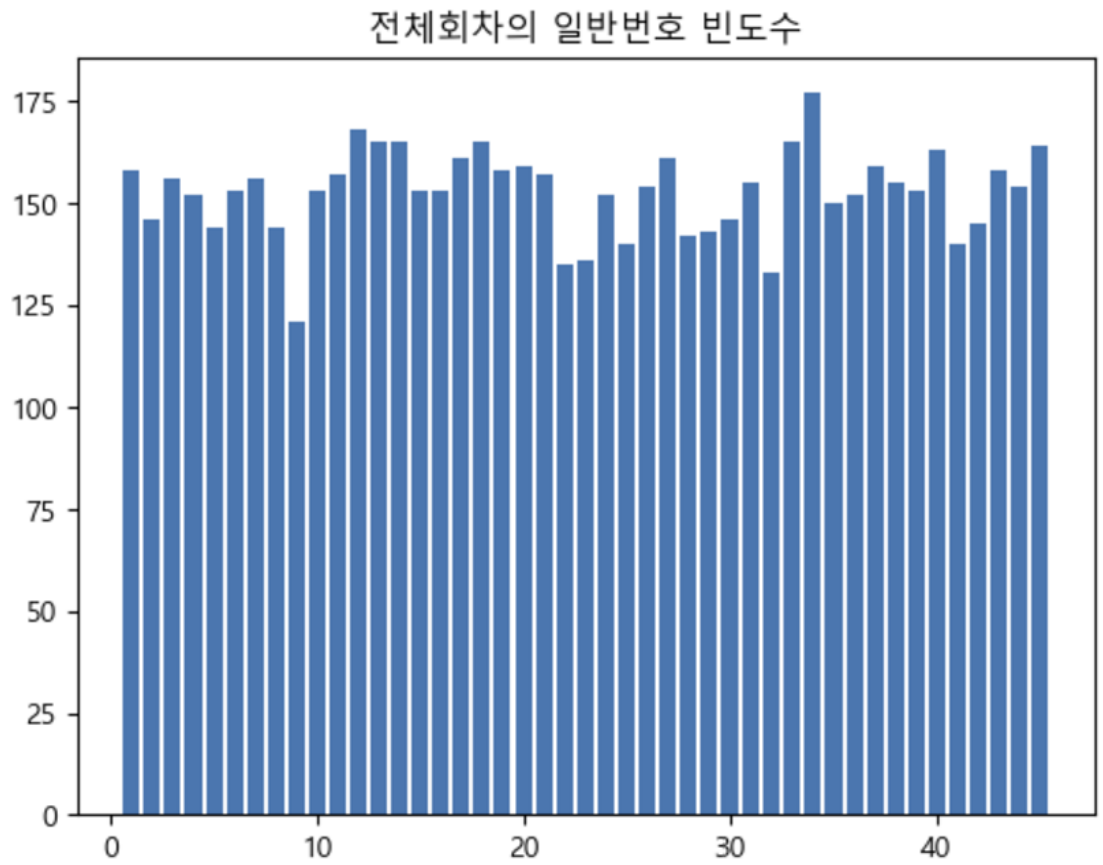
Value counts 함수를 사용하여
자료의 등장 횟수를 찾아
최빈값부터 차례대로 나열

```
In [9]: LDB_TC = LDB_T.apply(pd.Series.value_counts).sum(axis=1).sort_values(ascending=False)
        LDB_TC
```

```
Out[9]: Bonus
        43    35
        4     34
        6     33
        2     32
        32    32
        1     32
        30    31
        26    31
        33    30
        17    30
        24    30
        35    29
        27    29
        3     29
        38    28
        7     28
        10    27
        20    27
```

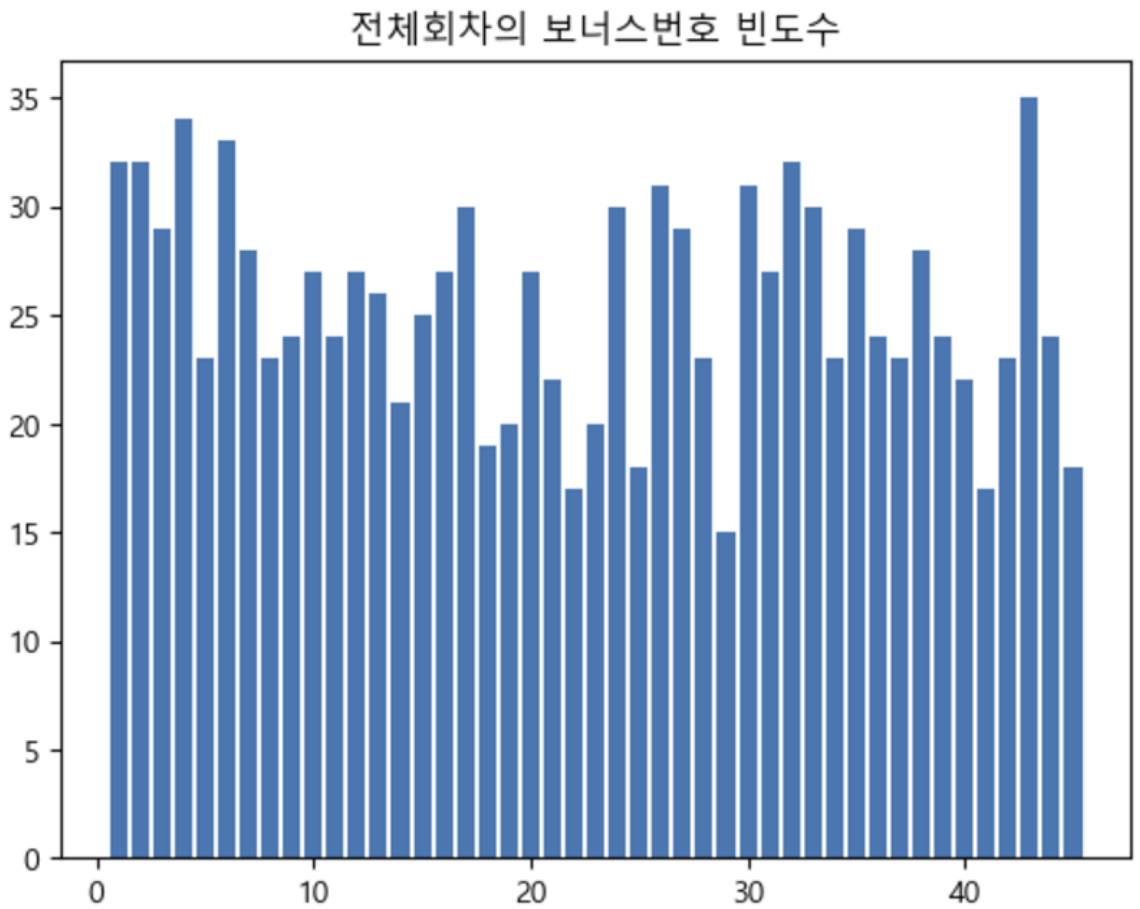
일반번호의 빈도수를 토대로 막대그래프를 통해 비교

```
In [10]: plt.title("전체회차의 일반번호 빈도수")
         plt.bar(LDN_TC.index, LDN_TC.values)
         plt.show()
```



보너스번호의 빈도수를 토대로 막대그래프를 통해 비교

In [11]:
```
plt.title("전체회차의 보너스번호 빈도수")
plt.bar(LDB_TC.index, LDB_TC.values)
plt.show()
```



In [12]:
```
#최근 1년간 추이
#52회차로 끊기

LDN_1Y = LDN_T.iloc[-52:]
LDB_1Y = LDB_T.iloc[-52:]
```

In [13]:
```
LDN_1YC = LDN_1Y.apply(pd.Series.value_counts).sum(axis=1).sort_values(ascending=False)
LDN_1YC
```

Out[13]:
```
33    12.0
34    12.0
19    12.0
30    11.0
40    11.0
13    11.0
16    10.0
28    10.0
6     10.0
12    10.0
37     9.0
38     9.0
21     9.0
31     9.0
14     9.0
7      9.0
3      9.0
11     8.0
44     8.0
```

In [14]:
```
LDB_1YC = LDB_1Y.apply(pd.Series.value_counts).sum(axis=1).sort_values(ascending=False)
LDB_1YC
```

Out[14]:
```
Bonus
4     4
8     3
2     3
22    3
3     3
32    3
12    2
6     2
20    2
24    2
15    2
34    2
26    2
28    2
23    2
7     2
33    1
18    1
```

최근 1년간
로또회차를 비교하기 위해
1년을 52주로 계산하여
lloc를 사용하여
52회차 추출

마찬가지로
1년간 로또번호에서도
일반번호와 보너스번호에서의
번호 빈도수 파악

In [15]:
```
#최근 5년간 추이
#220회차로 끊기

LDN_5Y = LDN_T.iloc[-220:]
LDB_5Y = LDB_T.iloc[-220:]
```

In [16]:
```
LDN_5YC = LDN_5Y.apply(pd.Series.value_counts).sum(axis=1).sort_values(ascending=False)
LDN_5YC
```

Out[16]:

35	37.0
12	36.0
33	36.0
6	36.0
30	35.0
13	35.0
45	34.0
14	34.0
3	33.0
21	33.0
7	33.0
16	32.0
37	32.0
38	32.0
29	32.0
44	32.0
15	31.0
40	31.0
11	31.0

In [17]:
```
LDB_5YC = LDB_5Y.apply(pd.Series.value_counts).sum(axis=1).sort_values(ascending=False)
LDB_5YC
```

Out[17]:

Bonus	
32	10
15	9
26	9
28	9
5	7
44	7
24	7
7	7
34	7
4	7
36	6
40	6
30	6
2	6
6	6
3	6
10	5
18	5

최근 5년간
로또회차를 비교하기 위해
5년을 220주로 계산하여
lloc를 사용하여
220회차 추출

마찬가지로
5년간 로또번호에서도
일반번호와 보너스번호에서의
번호 빈도수 파악

In [18]:
```
LDN_TCL = LDN_TC.index
LDN_TCL = LDN_TCL.tolist()
LDB_TCL = LDB_TC.index
LDB_TCL = LDB_TCL.tolist()
LDN_1YCL = LDN_1YC.index
LDN_1YCL = LDN_1YCL.tolist()
LDB_1YCL = LDB_1YC.index
LDB_1YCL = LDB_1YCL.tolist()
LDN_5YCL = LDN_5YC.index
LDN_5YCL = LDN_5YCL.tolist()
LDB_5YCL = LDB_5YC.index
LDB_5YCL = LDB_5YCL.tolist()
```

일반번호와 보너스번호에 대한 데이터프레임을
각각 전체, 1년, 5년에 대하여

Index 함수를 사용하여 특정 행 및 열 위치에 있는 값을 검색하고 반환

Tolist 함수를 사용하여 데이터프레임을 리스트로 변환

로또 번호 랜덤 출력 시스템 만들기

로또 변화 출력 시스템 시안

1) 모든 번호가 랜덤
2) 일반 번호 6개 중 3개는 전체회차에서 많이 나온 10개를 뽑아 그 중 3개를 랜덤 + 나머지 3개는 그냥 랜덤
3) 일반 번호 6개 중 3개는 전체회차에서 적게 나온 10개를 뽑아 그 중 3개를 랜덤 + 나머지 3개는 그냥 랜덤
4) 일반 번호 6개 중 3개는 최근 1년간 회차에서 많이 나온 10개를 뽑아 그 중 3개를 랜덤 + 나머지 3개는 그냥 랜덤
5) 일반 번호 6개 중 3개는 최근 1년간 회차에서 적게 나온 10개를 뽑아 그 중 3개를 랜덤 + 나머지 3개는 그냥 랜덤

In [19]:
```python
import random

#올랜덤

a = []
bonus = []

while len(a) < 6:
    b = random.randint(1,45)
    if b not in a:
        a.append(b)

while True:
    b = random.randint(1,45)
    if b not in a:
        bonus.append(b)
        break

print(f"{a} {bonus}")
```

```
[11, 31, 12, 13, 2, 37] [43]
```

랜덤함수를 불러오고
While 함수를 사용하여
번호가 6개 출력되면
함수 출력이 멈추도록 함

또 다른 while 함수를 사용하여
중복되는 값이 없을 때만
리스트 안에 값을 넣도록 함

In [20]:
```python
import random

def generate_lotto_number_1():
    main_numbers = []
    bonus_number = []

    while len(main_numbers) < 6:
        number = random.randint(1, 45)
        if number not in main_numbers:
            main_numbers.append(number)

    while True:
        number = random.randint(1, 45)
        if number not in main_numbers:
            bonus_number.append(number)
            break

    return main_numbers, bonus_number

main_numbers, bonus_number = generate_lotto_number_1()
print(f"Main numbers: {main_numbers}, Bonus number: {bonus_number}")
```

```
Main numbers: [39, 7, 29, 5, 20, 23], Bonus number: [14]
```

In [21]:
```
#전체회차에서 많이 나온 10개 중 3개 + 랜덤 3개 + 보너스

a = []
bonus = []

while len(a) < 6:
    if len(a) < 3:
        b = random.randint(0,9)
        if LDN_TCL[b] not in a:
            a.append(LDN_TCL[b])
    else:
        b = random.randint(1,45)
        if b not in a:
            a.append(b)

while True:
    b = random.randint(0,9)
    if LDB_TCL[b] not in a:
        bonus.append(LDB_TCL[b])
        break

print(f"{a} {bonus}")
```

```
[34, 18, 12, 25, 9, 14] [30]
```

랜덤함수를 불러오고
While 함수를 사용하여
숫자를 가장 많은 10개로 지정함

다른 while 함수를 이용하여
총 6개까지 추출하되
3개는 정의한 b에서 추출하게 함
나머지 3개는 랜덤하게
중복이 아니라면 추출되게 함

In [22]:
```
import random

def generate_lotto_number_2():
    main_numbers = []
    bonus_number = []

    while len(main_numbers) < 6:
        if len(main_numbers) < 3:
            number = random.randint(0,9)
            if LDN_TCL[number] not in main_numbers:
                main_numbers.append(LDN_TCL[number])
        else:
            number = random.randint(1, 45)
            if number not in main_numbers:
                main_numbers.append(number)

    while True:
        number = random.randint(0,9)
        if LDB_TCL[number] not in main_numbers:
            bonus_number.append(LDB_TCL[number])
            break

    return main_numbers, bonus_number

main_numbers, bonus_number = generate_lotto_number_2()
print(f"Main numbers: {main_numbers}, Bonus number: {bonus_number}")
```

```
Main numbers: [27, 33, 12, 6, 42, 35], Bonus number: [1]
```

In [23]:
```python
#전체회차에서 적게 나온 10개 중 3개 + 랜덤 3개 + 보너스

a = []
bonus = []

while len(a) < 6:
    if len(a) < 3:
        b = random.randint(-10,-1)
        if LDN_TCL[b] not in a:
            a.append(LDN_TCL[b])
    else:
        b = random.randint(1,45)
        if b not in a:
            a.append(b)

while True:
    b = random.randint(-10,-1)
    if LDB_TCL[b] not in a:
        bonus.append(LDB_TCL[b])
        break

print(f"{a} {bonus}")
```

```
[9, 22, 41, 43, 16, 23] [25]
```

랜덤함수를 불러오고
While 함수를 사용하여
숫자를 가장 적은 10개로 지정함

다른 while 함수를 이용하여
총 6개까지 추출하되
3개는 정의한 b에서 추출하게 함
나머지 3개는 랜덤하게
중복이 아니라면 추출되게 함

In [24]:
```python
import random

def generate_lotto_number_3():
    main_numbers = []
    bonus_number = []

    while len(main_numbers) < 6:
        if len(main_numbers) < 3:
            number = random.randint(-10, -1)
            if LDN_TCL[number] not in main_numbers:
                main_numbers.append(LDN_TCL[number])
        else:
            number = random.randint(1, 45)
            if number not in main_numbers:
                main_numbers.append(number)

    while True:
        number = random.randint(-10, -1)
        if LDB_TCL[number] not in main_numbers:
            bonus_number.append(LDB_TCL[number])
            break

    return main_numbers, bonus_number

main_numbers, bonus_number = generate_lotto_number_3()
print(f"Main numbers: {main_numbers}, Bonus number: {bonus_number}")
```

```
Main numbers: [32, 22, 9, 36, 10, 34], Bonus number: [18]
```

랜덤함수를 불러오고
While 함수를 사용하여
숫자를 최근 1년 중 가장 많은 10개로 지정함

다른 while 함수를 이용하여
총 6개까지 추출하되
3개는 정의한 b에서 추출하게 함
나머지 3개는 랜덤하게
중복이 아니라면 추출되게 함

In [25]:
```
#최근 1년간 많이 나온 수 10개 중 3개 + 랜덤 3개 + 보너스

a = []
bonus = []

while len(a) < 6:
    if len(a) < 3:
        b = random.randint(0,9)
        if LDN_1YCL[b] not in a:
            a.append(LDN_1YCL[b])
    else:
        b = random.randint(1,45)
        if b not in a:
            a.append(b)

while True:
    b = random.randint(0,9)
    if LDB_1YCL[b] not in a:
        bonus.append(LDB_1YCL[b])
        break

print(f"{a} {bonus}")
```

```
[13, 6, 7, 17, 35, 18] [8]
```

In [26]:
```
import random

def generate_lotto_number_4():
    main_numbers = []
    bonus_number = []

    while len(main_numbers) < 6:
        if len(main_numbers) < 3:
            number = random.randint(0,9)
            if LDN_1YCL[number] not in main_numbers:
                main_numbers.append(LDN_1YCL[number])
        else:
            number = random.randint(1, 45)
            if number not in main_numbers:
                main_numbers.append(number)

    while True:
        number = random.randint(0,9)
        if LDB_1YCL[number] not in main_numbers:
            bonus_number.append(LDB_1YCL[number])
            break

    return main_numbers, bonus_number

main_numbers, bonus_number = generate_lotto_number_4()
print(f"Main numbers: {main_numbers}, Bonus number: {bonus_number}")
```

```
Main numbers: [30, 7, 12, 13, 15, 19], Bonus number: [22]
```

In [27]:
```
#최근 1년간 적게 나온 수 10개 중 3개 + 랜덤 3개 + 보너스

a = []
bonus = []

while len(a) < 6:
    if len(a) < 3:
        b = random.randint(-10,-1)
        if LDN_1YCL[b] not in a:
            a.append(LDN_1YCL[b])
    else:
        b = random.randint(1,45)
        if b not in a:
            a.append(b)

while True:
    b = random.randint(-10,-1)
    if LDB_1YCL[b] not in a:
        bonus.append(LDB_1YCL[b])
        break

print(f"{a} {bonus}")
```

```
[25, 42, 8, 44, 5, 19] [30]
```

랜덤함수를 불러오고
While 함수를 사용하여
숫자를 최근 1년 중 가장 적은 10개로 지정함

다른 while 함수를 이용하여
총 6개까지 추출하되
3개는 정의한 b에서 추출하게 함
나머지 3개는 랜덤하게
중복이 아니라면 추출되게 함

In [28]:
```
import random

def generate_lotto_number_5():
    main_numbers = []
    bonus_number = []

    while len(main_numbers) < 6:
        if len(main_numbers) < 3:
            number = random.randint(-10, -1)
            if LDN_1YCL[number] not in main_numbers:
                main_numbers.append(LDN_1YCL[number])
        else:
            number = random.randint(1, 45)
            if number not in main_numbers:
                main_numbers.append(number)

    while True:
        number = random.randint(-10, -1)
        if LDB_1YCL[number] not in main_numbers:
            bonus_number.append(LDB_1YCL[number])
            break

    return main_numbers, bonus_number

main_numbers, bonus_number = generate_lotto_number_5()
print(f"Main numbers: {main_numbers}, Bonus number: {bonus_number}")
```

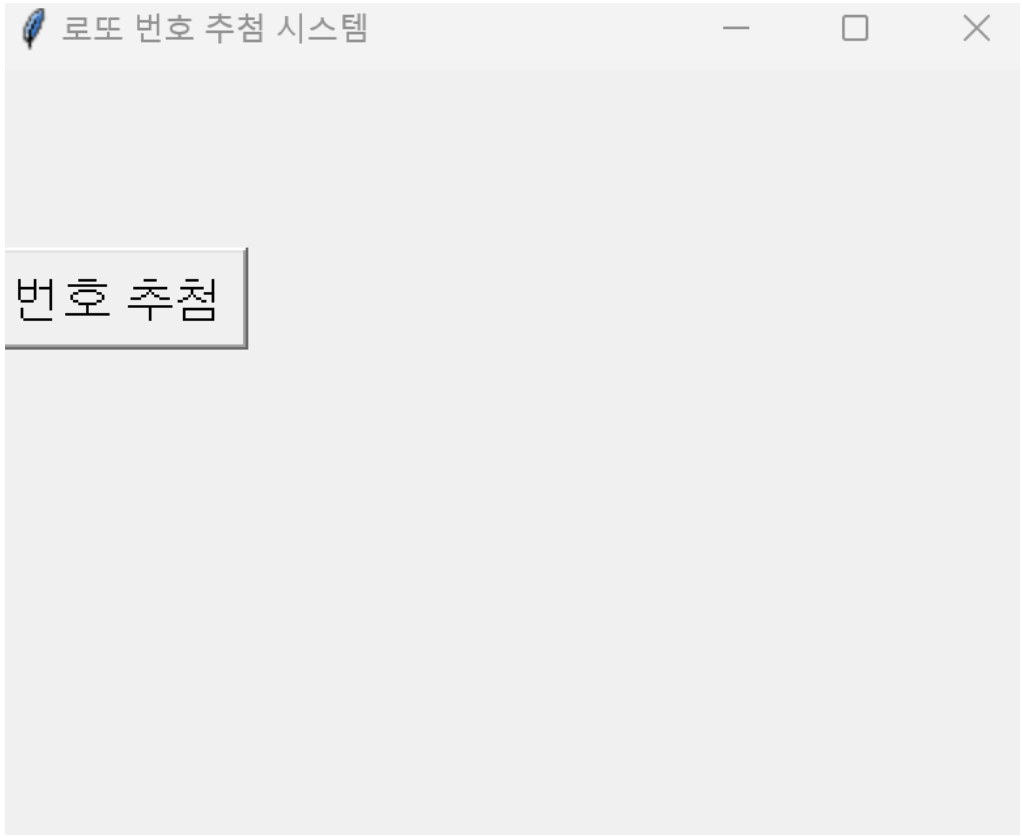
```
Main numbers: [8, 2, 39, 34, 23, 28], Bonus number: [18]
```

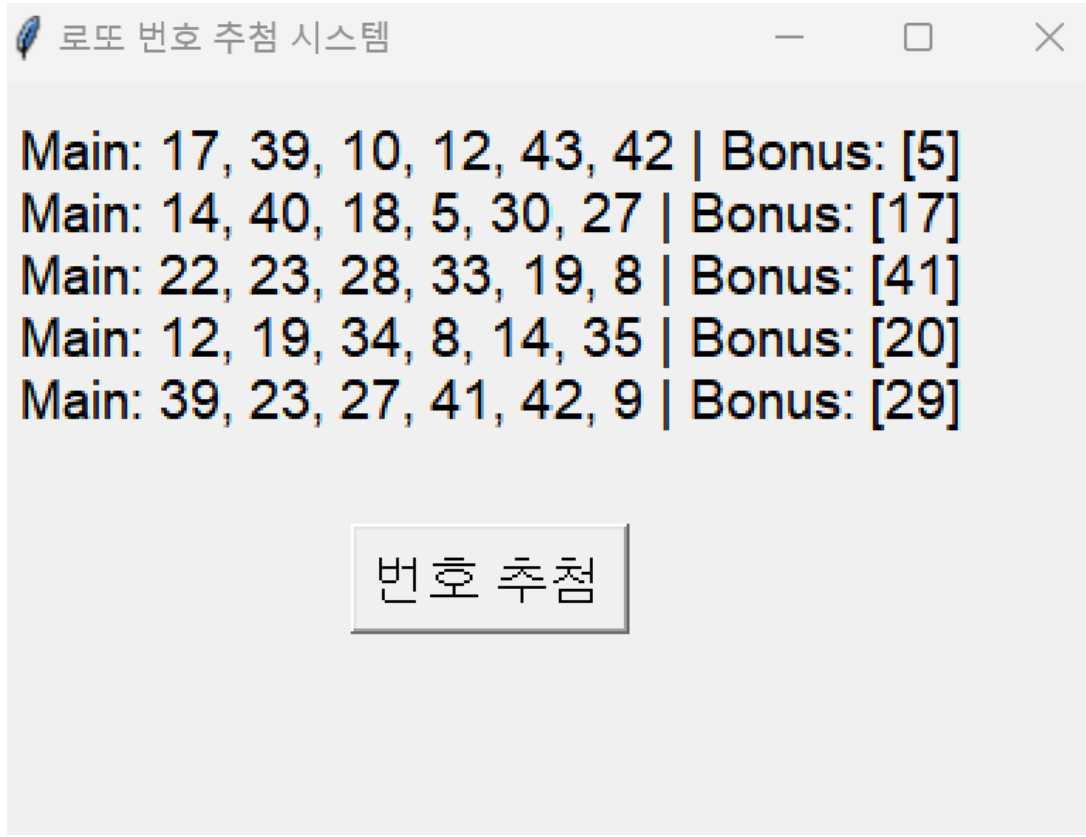
```
root = tk.Tk()
root.title("로또 번호 추첨 시스템")
root.geometry("400x300")

result_label = tk.Label(root, text="", font=("Helvetica", 14), justify="left", anchor="w")
result_label.grid(row=0, column=0, columnspan=5, padx=10, pady=10)

def draw_numbers():
    functions = [
        generate_lotto_number_1,
        generate_lotto_number_2,
        generate_lotto_number_3,
        generate_lotto_number_4,
        generate_lotto_number_5,
    ]
    results = [func() for func in functions]

    display_text = "₩n".join(
        [f"Main: {', '.join(map(str, main))} | Bonus: {bonus}" for main, bonus in results]
    )

    result_label.config(text=display_text)

draw_button = tk.Button(root, text="번호 추첨", command=draw_numbers, font=("Helvetica", 14))
draw_button.grid(row=1, column=0, columnspan=5, pady=20)

root.mainloop()
```

앞에서 만든 함수 5개를 모두 사용하는
함수 draw_numbers를 생성

각 수가 일반번호와 보너스번호로
분리되어 나오도록 함

버튼을 만들어서
번호추첨이라는 버튼을 누르면 번호가 추첨되도록 함





1147회 당첨결과

(2024년 11월 23일 추첨)

7 11 24 26 27 37 + 32

당첨번호 보너스