

201944036 김주연

흥국생명 배구팀 분석

목차

1 데이터 수집

2 데이터 저장

3 데이터 가공/정제

4 데이터 분석

데이터 수집 - 크롤링 사이트

도드람 2021-2022 V-리그

NH농협 2009-2010 V-리그

NH농협 2008-2009 V-리그

NH농협 2007-2008 V-리그

HILLSTATE 2006-2007 V-리그

KT&G 2005-2006 V-리그

KT&G 2005 V-리그

1. 시즌 순회

2. 경기날짜 순회

2021년 10월

라운드 선택

이동

| 경기 | 시간 | 중계 | 라운드 | 비고 | | | | |
|-----------|-------|-----------|--------------------|------------------------|--------|-------------|------------|-----------------|
| 3:1 우리카드 | 14:00 | Sports | V리그 1Round | 하이라이트 경기요약 상세결과 | | | | |
| 3:0 흥국생명 | 16:00 | Sports | V리그 1Round | 하이라이트 경기요약 상세결과 | | | | |
| 10.17 (일) | 3 | 남자 | 현대캐피탈 3:1 OK금융그룹 | 17:00 | | | | |
| | 4 | 여자 | 현대건설 3:1 IBK기업은행 | 14:33 | 수원 | KBS1 | V리그 1Round | 하이라이트 경기요약 상세결과 |
| 10.18 (월) | | 경기가 없습니다. | | | | | | |
| 10.19 (화) | 5 | 남자 | 삼성화재 0:3 한국전력 | 19:00 | 대전충무 | | V리그 1Round | 하이라이트 경기요약 상세결과 |
| | 6 | 여자 | 페퍼저축은행 1:3 KGC인삼공사 | 19:00 | 페퍼스타디움 | KBSN Sports | V리그 1Round | 하이라이트 경기요약 상세결과 |
| 10.20 (수) | 7 | 남자 | KB손해보험 3:2 현대캐피탈 | 19:00 | 의정부 | | V리그 1Round | 하이라이트 경기요약 상세결과 |
| | 8 | 여자 | 한국도로공사 0:3 현대건설 | 19:00 | 김천실내 | KBSN Sports | V리그 1Round | 하이라이트 경기요약 상세결과 |
| 10.21 (목) | 9 | 남자 | OK금융그룹 3:2 우리카드 | 19:00 | 안산상록수 | | V리그 1Round | 하이라이트 경기요약 상세결과 |

3. 여자 경기 중 흥국생명의 경기만 상세결과 버튼 클릭

데이터 수집 – 크롤링 사이트

상세결과 페이지에서 상대팀 데이터 일부와 흥국생명의 경기 데이터 가져오기

도드람 2021-2022 V-리그

상세결과

경기일정 돌아가기

2021년 10월 16일 (토) / 제2경기

| | | | | | | | | | | |
|---|--------------------------|-------|------|------|------|------|------|-------|-------------|--------------|
| WIN | SET SCORE | 여자부 | 1SET | 2SET | 3SET | 4SET | 5SET | TOTAL | SET SCORE | LOSE |
| GS칼텍스 | 3 | GS칼텍스 | 25 | 27 | 25 | 0 | 0 | 77 | 0 | PINK SPIDERS |
| | | 흥국생명 | 21 | 25 | 22 | 0 | 0 | 68 | | |
| | | 경기시간 | 26 m | 34 m | 29 m | 0 m | 0 m | 1h29m | | |
| 3위(20승 11패) | 서울장충체육관 / 16:00 / 관중수 0명 | | | | | | | | 6위(10승 23패) | |
| [심판진] 경기위원: 김태중 / 심판위원: 정유연 / 주심: 남영수 / 부심: 최성권 | | | | | | | | | | |

팀기록

선수기록

문자중계

사진 및 영상

기록비교

GS칼텍스

흥국생명

공격득점

52

38

블로킹득점

5

10

서브득점

5

3

상대범실

15

17

전체득점

77

68

디그성공

64

57

리시브정확

37

20

세트성공

50

34

| 공격 성공률 | 리시브효율 |
|-----------------------|-------------------|
| 42.98% | 51.52% |
| 36.89% | 22.73% |
| 공격점유율 TOP3와 성공률 | |
| 점유율(%) / 성공률(%) | |
| 모마 37.19 / 42.22 | 캣벨 46.6 / 35.42 |
| 유서연 22.31 / 44.44 | 김미연 20.39 / 38.1 |
| 강소휘 20.66 / 44 | 김다은 13.59 / 35.71 |
| 세트당 수비 TOP3와 리시브 효율 | |
| 세트당 수비(개) / 리시브 효율(%) | |
| 유서연 4.67 / 46.15 | 김해란 4.33 / 33.33 |
| 강소휘 3.67 / 45.16 | 캣벨 4 / 0 |
| 모마 3.33 / 0 | 김미연 3.33 / 31.25 |

가져오는 데이터: 상대팀, 공격득점, 블로킹득점, 서브득점, 팀범실, 상대범실, 전체득점, 상대득점, 디그성공, 리시브정확, 세트성공, 공격성공률, 리시브효율, 세트승, 세트패, 결과

데이터 수집 – 크롤링 사이트

김연경 선수가 출전했으면 김연경 선수의 데이터 가져오기

도드람 2020-2021 V-리그

상세결과

경기일정 돌아가기

2020년 10월 21일 (수) / 제 7경기

| | | | | | | | | | | |
|---|--------------------------|-------|------|------|------|------|------|-------|-----------|-----------|
| LOSE | SET SCORE | 여자부 | 1SET | 2SET | 3SET | 4SET | 5SET | TOTAL | SET SCORE | WIN |
| | 1 | GS칼텍스 | 27 | 28 | 28 | 17 | 0 | 100 | 3 | |
| | | 홍국생명 | 29 | 30 | 26 | 25 | 0 | 110 | | |
| | | 경기시간 | 34 m | 36 m | 34 m | 29 m | 0 m | 2h13m | | |
| 1위(3승 0패) | 서울장충체육관 / 15:30 / 관중수 0명 | | | | | | | | | 1위(2승 1패) |
| [심판진] 경기위원: 유애자 / 심판위원: 김영철 / 주심: 전영아 / 부심: 권대진 | | | | | | | | | | |

팀기록

선수기록

문자중계

사진 및 영상

달성기록

| | |
|--|-----------------|
| | |
| 팀 | 홍국생명 |
| 역대통산 디그성공 36000개 돌파(4호) 현재36013개, 오늘 110개 | 달성기록이 없습니다. |

선수별 상세기록

GS칼텍스

홍국생명

홍국생명핑크스파이더스

< 이전기록 1/5 다음기록 >

| No. | 이름 | 출전세트 | | | | | 득점 | 공격종합 | | | | | |
|-----|----------|------|------|------|------|------|----|------|----|------|----|--------|-------|
| | | 1set | 2set | 3set | 4set | 5set | | 시도 | 성공 | 공격차단 | 범실 | 성공률 | 점유율 |
| 12 | 루시아 (R) | ■ | ■ | ■ | ■ | | 27 | 55 | 24 | 1 | 2 | 43.64 | 32.54 |
| 10 | 김연경 (L) | ■ | ■ | ■ | ■ | | 25 | 47 | 20 | 3 | 4 | 42.55 | 27.81 |
| 17 | 이재영 (L) | ■ | ■ | ■ | ■ | | 19 | 51 | 18 | 4 | 2 | 35.29 | 30.18 |
| 13 | 김세영 (C) | ■ | ■ | ■ | ■ | | 10 | 6 | 5 | 0 | 0 | 83.33 | 3.55 |
| 11 | 이다영 (S) | ■ | ■ | ■ | ■ | | 5 | 5 | 3 | 0 | 0 | 60.00 | 2.96 |
| 15 | 김채연 (C) | 4 | ■ | ■ | ■ | | 3 | 4 | 1 | 1 | 0 | 25.00 | 2.37 |
| 4 | 이주아 (C) | ■ | | | | | 1 | 1 | 1 | 0 | 0 | 100.00 | 0.59 |
| 19 | 김미연 (L) | | 12 | 12 | 15 | | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 8 | 도수빈 (Li) | Li | Li | Li | Li | | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 5 | 박상미 (Li) | Li | Li | Li | Li | | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 14 | 박현주 (L) | 13 | 13 | 13 | 13 | | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |
| 16 | 이한비 (L) | | | 15 | | | 0 | 0 | 0 | 0 | 0 | 0.00 | 0.00 |

가져오는 데이터: 김연경선수 출전여부, 김연경시도, 김연경성공, 김연경공격차단, 김연경범실, 김연경성공률, 김연경점유율

데이터 수집 - 크롤링 코드

```
try:
    for i in range(1, 19):
        if i < 10:
            page_name = 'https://www.kovo.co.kr/game/v-league/11110_schedule_list.asp?season=00' + str(i)
        else:
            page_name = 'https://www.kovo.co.kr/game/v-league/11110_schedule_list.asp?season=0' + str(i)

        driver.get(page_name)
```

각 시즌을 모두 순회하며 페이지 불러오기

```
select = Select(driver.find_element(By.CSS_SELECTOR, "#wrap_content > article > div.wrp_date > "
                                     "form > fieldset > select.selectbox_table.w120.selectBox"))

for j in range(2, len(select.options) + 1):
    select_option = driver.find_element(By.CSS_SELECTOR, "#wrap_content > article > div.wrp_date > form > fieldset > "
                                                         "select.selectbox_table.w120.selectBox > option:nth-child("
                                                         + str(j) + ")").get_attribute("value")
    page_name_new = page_name + "&team=&yymm=" + select_option + "&r_round="
    driver.get(page_name_new)
```

각 시즌의 페이지에서 날짜 선택하며 페이지 불러오기

데이터 수집 - 크롤링 코드

```
table = driver.find_element(By.XPATH, "//*[@id='type1']/div/table/tbody")
tr = table.find_elements(By.TAG_NAME, "tr")

for k in range(1, len(tr) + 1):
    if driver.find_element(By.CSS_SELECTOR, "#type1 > div > table > tbody > tr:nth-child(" + str(k) +
        ") > td:nth-child(3)").size.get("width") < 40 :
        gender = driver.find_element(By.CSS_SELECTOR, "#type1 > div > table > tbody > tr:nth-child("
            + str(k) + ") > td:nth-child(3)").get_attribute("innerText")

        if (gender == "여자"):
            left = driver.find_element(By.CSS_SELECTOR, "#type1 > div > table > tbody > tr:nth-child("
                + str(k) + ") > td.tleft").get_attribute("innerText")
            right = driver.find_element(By.CSS_SELECTOR, "#type1 > div > table > tbody > tr:nth-child("
                + str(k) + ") > td.tright").get_attribute("innerText")
```

각 페이지의 테이블을 순회하며 여자부 경기인지
판단하여 여자경기이면 흥국생명의 경기인지 확인

```
# 흥국생명의 경기인지 판단
if "흥국생명" in left or "흥국생명" in right:
    driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/div/a").send_keys(Keys.ENTER)
# 상세결과의 데이터 가져오기
if i < 10:
    driver.find_element(By.CSS_SELECTOR, "#type1 > div > table > tbody > tr:nth-child(" + str(k) +
        ") > td:nth-child(10) > a").click()
else:
    driver.find_element(By.CSS_SELECTOR, "#type1 > div > table > tbody > tr:nth-child(" + str(k) +
        ") > td:nth-child(10) > a.btn.btn_lst.wrp_rounded.w82.btn_grey").click()
```

흥국생명의 경기면 상세결과 페이지 클릭

데이터 수집 - 크롤링 코드

```
if check_element("//*[@id='wrp_content']/article[1]/table/tbody/tr/td[3]/table/tbody/tr[3]/td[7]") :  
  
    game_time_array = driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/table/"  
                                                "tbody/tr/td[3]/table/tbody/tr[3]/td[7]")\  
        .get_attribute("innerHTML").rstrip("m").split("h")  
    game_time = 60 * int(game_time_array[0]) + int(game_time_array[1])  
    h_team = driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/table/tbody/tr/td[1]/p[2]/span[2]").get_attribute("innerHTML")  
    h_score = driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/table/tbody/tr/td[2]/p[2]").get_attribute("innerHTML")  
    h_attack = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[1]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_block = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[2]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_serve = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[3]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_miss = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[4]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_all = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[5]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_dig = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[6]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_recieve = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[7]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_set = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[8]/dd[1]/div[1]/span").get_attribute("innerHTML")  
    h_attackpercent = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[2]/div/div[1]/div[1]/div[1]/span/span/span").get_attribute("innerHTML").rstrip("%")  
    h_recievepercent = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[2]/div/div[1]/div[2]/div[1]/span/span/span").get_attribute("innerHTML").rstrip("%")  
    a_team = driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/table/tbody/tr/td[5]/p[2]/span[2]").get_attribute("innerHTML")  
    a_score = driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/table/tbody/tr/td[4]/p[2]").get_attribute("innerHTML")  
    a_attack = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[1]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_block = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[2]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_serve = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[3]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_miss = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[4]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_all = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[5]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_dig = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[6]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_recieve = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[7]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_set = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[1]/div/div/dl[8]/dd[1]/div[2]/span").get_attribute("innerHTML")  
    a_attackpercent = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[2]/div/div[1]/div[1]/div[2]/span/span/span").get_attribute("innerHTML").rstrip("%")  
    a_recievepercent = driver.find_element(By.XPATH, "//*[@id='tab1']/div[1]/div[2]/div/div[1]/div[2]/div[2]/span/span/span").get_attribute("innerHTML").rstrip("%")
```

상세페이지의 경기결과 가져오기
흥국생명의 결과와 상대팀의 결과 둘 다 가져오기

데이터 수집 - 크롤링 코드

```
# 선수 기록 페이지
driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[2]/ul/li[2]/a").send_keys(Keys.ENTER)

if "홍국생명" in h_team:
    if check_element("//*[@id='tab2']/h4[2]") :
        if check_element("//*[@id='tab2']/div[3]/ul/li[1]/a") :
            driver.find_element(By.XPATH, "//*[@id='tab2']/div[3]/ul/li[1]/a").send_keys(Keys.ENTER)
        else :
            driver.find_element(By.XPATH, "//*[@id='tab2']/div[2]/ul/li[1]/a").send_keys(Keys.ENTER)
    else :
        driver.find_element(By.XPATH, "//*[@id='tab2']/div[1]/ul/li[1]/a").send_keys(Keys.ENTER)
    team = "team1"
else:
    if check_element("//*[@id='tab2']/h4[2]") :
        if check_element("//*[@id='tab2']/div[3]/ul/li[1]/a"):
            driver.find_element(By.XPATH, "//*[@id='tab2']/div[3]/ul/li[2]/a").send_keys(Keys.ENTER)
        else:
            driver.find_element(By.XPATH, "//*[@id='tab2']/div[2]/ul/li[2]/a").send_keys(Keys.ENTER)
    else:
        driver.find_element(By.XPATH, "//*[@id='tab2']/div[1]/ul/li[2]/a").send_keys(Keys.ENTER)
    team = "team2"
```

상세 결과 페이지에서 선수 기록 클릭

데이터 수집 - 크롤링 코드

```
table = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[1]/tbody")
tr = table.find_elements(By.TAG_NAME, "tr")
kim_list = []

for n in range(1, len(tr) + 1):
    kim_check = driver.find_element(By.CSS_SELECTOR, "#" + team +
                                     " > div > div.wrp_lst > table.lst_board.lst_fixed.w123 > tbody "
                                     "> tr:nth-child(" + str(n) + ") > td.name").get_attribute("innerText")
    kim_list.append(kim_check.split(" ")[0])
```

선수 기록 테이블 순회하면서 김연경 선수의 기록이 있는지 확인

데이터 수집 - 크롤링 코드

```
if "김연경" in kim_list:
    num = kim_list.index("김연경")
    for m in range(1, 6):
        starting_check = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr[1]/td["
                                                + str(m) + "]/span").get_attribute("class")
        if "starting" in starting_check or "switch" in starting_check:
            starting = "선발"
            kim_score = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[6]").get_attribute("innerText")
            kim_trial = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[7]").get_attribute("innerText")
            kim_success = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[8]").get_attribute("innerText")
            kim_block = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[9]").get_attribute("innerText")
            kim_miss = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[10]").get_attribute("innerText")
            kim_successpercent = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[11]").get_attribute("innerText")
            kim_share = driver.find_element(By.XPATH, "//*[@id='" + team + "']/div/div[2]/table[2]/tbody/tr["
                                                + str(num + 1) + "]/td[12]").get_attribute("innerText")
            break;
```

김연경 선수의 기록이 있으면 기록 가져오기

데이터 수집 - 크롤링 코드

```
        else:
            starting = "미출전"
            kim_score = ""
            kim_trial = ""
            kim_success = ""
            kim_block = ""
            kim_miss = ""
            kim_successpercent = ""
            kim_share = ""
            break;
    else:
        starting = "미출전"
        kim_score = ""
        kim_trial = ""
        kim_success = ""
        kim_block = ""
        kim_miss = ""
        kim_successpercent = ""
        kim_share = ""
```

김연경 선수의 기록이 없으면 기록 비워두기

데이터 수집 – 데이터 표

| | |
|----|-------|
| 0 | 시즌 |
| 1 | 상대팀 |
| 2 | 공격득점 |
| 3 | 블로킹득점 |
| 4 | 서브득점 |
| 5 | 팀범실 |
| 6 | 상대범실 |
| 7 | 전체득점 |
| 8 | 상대득점 |
| 9 | 디그성공 |
| 10 | 리시브정확 |
| 11 | 세트성공 |
| 12 | 공격성공률 |

| | |
|----|---------|
| 13 | 리시브효율 |
| 14 | 경기시간 |
| 15 | 세트승 |
| 16 | 세트패 |
| 17 | 결과 |
| 18 | 김연경 |
| 19 | 김연경시도 |
| 20 | 김연경성공 |
| 21 | 김연경공격차단 |
| 22 | 김연경범실 |
| 23 | 김연경성공률 |
| 24 | 김연경점유율 |

데이터 저장

```
f = open('volleyball.csv', 'w', newline='')
wr = csv.writer(f)
wr.writerow(['시즌', '상대팀', '공격득점', '블로킹득점', '서브득점', '팀범실', '상대범실',
            '전체득점', '상대득점', '디그성공', '리시브정확', '세트성공', '공격성공률',
            '리시브효율', '경기시간', '세트승', '세트패', '결과', '김연경', '김연경득점',
            '김연경시도', '김연경성공', '김연경공격차단', '김연경범실', '김연경성공률', '김연경점유율'])
```

파일 작성할 준비하고 헤더 저장

```
if "흥국생명" in h_team:
    result = '승' if int(h_score) > int(a_score) else '패'
    wr.writerow([i, a_team, h_attack, h_block, h_serve, a_miss, h_miss, h_all, a_all, h_dig, h_recieve, h_set,
                h_attackpercent, h_recievepercent, game_time, h_score, a_score, result, starting, kim_score,
                kim_trial, kim_success, kim_block, kim_miss, kim_successpercent, kim_share])
elif "흥국생명" in a_team:
    result = '승' if int(h_score) < int(a_score) else '패'
    wr.writerow([i, h_team, a_attack, a_block, a_serve, h_miss, a_miss, a_all, h_all, a_dig, a_recieve, a_set,
                a_attackpercent, a_recievepercent, game_time, a_score, h_score, result, starting, kim_score,
                kim_trial, kim_success, kim_block, kim_miss, kim_successpercent, kim_share])
driver.find_element(By.XPATH, "//*[@id='wrp_content']/article[1]/div/a").send_keys(Keys.ENTER)
else :
    driver.back()
```

가져온 기록을 csv파일로 저장

데이터 가공/정제 – 데이터 전처리

2. 데이터 전처리

1. 상대팀 column에서 KT&G, 인삼공사를 KGC인삼공사로 치환
2. 김연경 column에서 선발을 출전으로 치환

```
volleyball = pd.read_csv("/content/volleyball.csv", encoding="cp949")

# 1. 팀명
volleyball.replace({"KT&G": "KGC인삼공사", "인삼공사": "KGC인삼공사"}, inplace=True)

# 2. 출전 여부
volleyball.replace("선발", "출전", inplace=True)

volleyball.head(20)
```

| | 시즌 | 상대팀 | 공격 득점 | 블로킹 득점 | 서브 득점 | 팀범 실 | 상대 범실 | 전체득 점 | 상대득 점 | 디그성 공 | ... | 세트 패 | 결 과 | 김연 경 |
|---|----|---------|----------|-----------|----------|---------|----------|----------|----------|----------|-----|---------|--------|---------|
| 0 | 1 | 현대건설 | 43 | 3 | 5 | 16 | 18 | 75 | 66 | 55 | ... | 0 | 승 | 미출전 |
| 1 | 1 | KGC인삼공사 | 37 | 3 | 4 | 19 | 11 | 59 | 81 | 75 | ... | 3 | 패 | 미출전 |
| 2 | 1 | 도로공사 | 35 | 1 | 1 | 17 | 8 | 50 | 75 | 73 | ... | 3 | 패 | 미출전 |
| 3 | 1 | GS칼텍스 | 68 | 6 | 2 | 29 | 17 | 100 | 96 | 91 | ... | 1 | 승 | 미출전 |
| 4 | 1 | GS칼텍스 | 51 | 3 | 4 | 25 | 20 | 83 | 96 | 102 | ... | 3 | 패 | 미출전 |
| 5 | 1 | 도로공사 | 36 | 1 | 3 | 26 | 10 | 52 | 75 | 109 | ... | 3 | 패 | 미출전 |
| 6 | 1 | 현대건설 | 64 | 5 | 11 | 25 | 20 | 101 | 109 | 138 | ... | 3 | 패 | 미출전 |

팀명과 원소 일부 변경

데이터 분석

3. 데이터 살펴보기

- 1. 총 데이터 개수
- 2. 데이터 크기
- 3. 데이터 요약 정보
- 4. 데이터 통계 정보

```
# 1. 총 데이터 개수
print("1. 총 데이터 개수")
len(volleyball)
print()

# 2. 데이터 크기
print("2. 데이터 크기")
volleyball.shape
print()

# 3. 데이터 요약 정보
print("3. 데이터 요약 정보")
volleyball.info()
print()

# 4. 데이터 통계 정보
print("4. 데이터 통계 정보")
volleyball.describe()
```

- 1. 총 데이터 개수
- 2. 데이터 크기
- 3. 데이터 요약 정보

RangeIndex: 552 entries, 0 to 551
Data columns (total 26 columns):

| # | Column | Non-Null Count | Dtype |
|----|----------|----------------|---------|
| 0 | 시즌 | 552 non-null | int64 |
| 1 | 상대팀 | 552 non-null | object |
| 2 | 공격득점 | 552 non-null | int64 |
| 3 | 블로킹득점 | 552 non-null | int64 |
| 4 | 서브득점 | 552 non-null | int64 |
| 5 | 팀범실 | 552 non-null | int64 |
| 6 | 상대범실 | 552 non-null | int64 |
| 7 | 전체득점 | 552 non-null | int64 |
| 8 | 상대득점 | 552 non-null | int64 |
| 9 | 디그성공 | 552 non-null | int64 |
| 10 | 리시브정확 | 552 non-null | int64 |
| 11 | 세트성공 | 552 non-null | int64 |
| 12 | 공격성공률 | 552 non-null | float64 |
| 13 | 리시브효율 | 552 non-null | float64 |
| 14 | 경기시간 | 552 non-null | int64 |
| 15 | 세트승 | 552 non-null | int64 |
| 16 | 세트패 | 552 non-null | int64 |
| 17 | 결과 | 552 non-null | object |
| 18 | 김연경 | 552 non-null | object |
| 19 | 김연경 득점 | 162 non-null | float64 |
| 20 | 김연경 시도 | 162 non-null | float64 |
| 21 | 김연경 성공 | 162 non-null | float64 |
| 22 | 김연경 공격차단 | 162 non-null | float64 |
| 23 | 김연경 범실 | 162 non-null | float64 |
| 24 | 김연경 성공률 | 162 non-null | float64 |
| 25 | 김연경 점유율 | 162 non-null | float64 |

dtypes: float64(9), int64(14), object(3)
memory usage: 112.2+ KB

4. 데이터 통계 정보

| | 시즌 | 공격득점 | 블로킹득점 | 서브득점 | 팀범실 | 상대범실 | 전체득점 | 상대득점 | 디그성공 | 리시브정확 | ... |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-----|
| count | 552.000000 | 552.000000 | 552.000000 | 552.000000 | 552.000000 | 552.000000 | 552.000000 | 552.000000 | 552.000000 | 552.000000 | ... |
| mean | 9.802536 | 52.335145 | 6.791667 | 4.494565 | 21.061594 | 19.905797 | 84.916667 | 85.117754 | 81.255435 | 35.295290 | ... |
| std | 5.097146 | 12.388376 | 3.143005 | 2.358898 | 6.209867 | 6.013461 | 17.836023 | 17.278510 | 20.791899 | 11.620033 | ... |
| min | 1.000000 | 21.000000 | 0.000000 | 0.000000 | 7.000000 | 6.000000 | 40.000000 | 44.000000 | 27.000000 | 9.000000 | ... |
| 25% | 5.000000 | 43.750000 | 4.000000 | 3.000000 | 17.000000 | 16.000000 | 75.000000 | 75.000000 | 66.000000 | 27.000000 | ... |
| 50% | 10.000000 | 53.000000 | 7.000000 | 4.000000 | 21.000000 | 19.000000 | 87.000000 | 87.000000 | 80.000000 | 33.000000 | ... |
| 75% | 14.000000 | 61.000000 | 9.000000 | 6.000000 | 25.000000 | 24.000000 | 99.000000 | 99.000000 | 95.250000 | 44.000000 | ... |
| max | 18.000000 | 94.000000 | 19.000000 | 12.000000 | 38.000000 | 39.000000 | 125.000000 | 117.000000 | 156.000000 | 86.000000 | ... |

데이터 분석

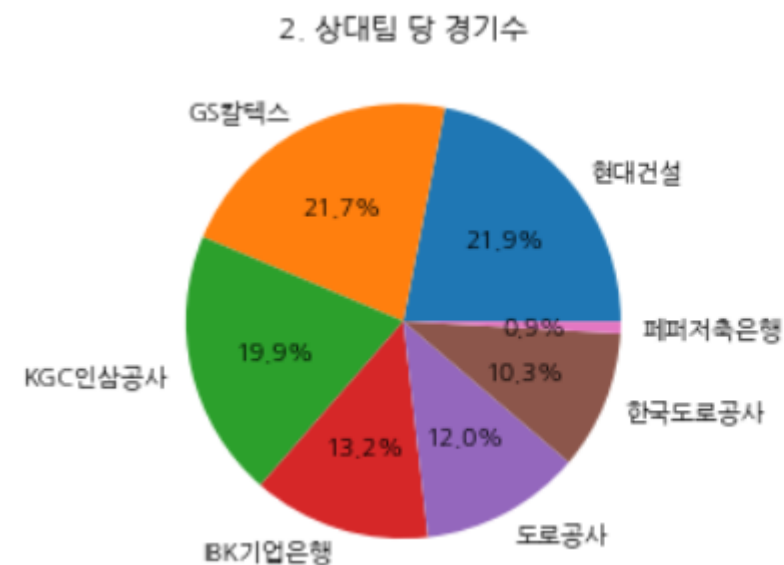
4. 시각화

1. 시즌 당 경기 수
2. 상대팀 당 경기 수
3. 승패

```
# 1. 시즌 당 경기수
season = []
game_count = []
season_data = volleyball['시즌'].value_counts().to_dict()
for k, v in season_data.items():
    season.append(k)
    game_count.append(v)
plt.subplot(1, 1, 1)
plt.pie(game_count, labels=season, autopct='%0.1f%%', pctdistance=0.85)
plt.title("1. 시즌 당 경기수")
plt.show()
```

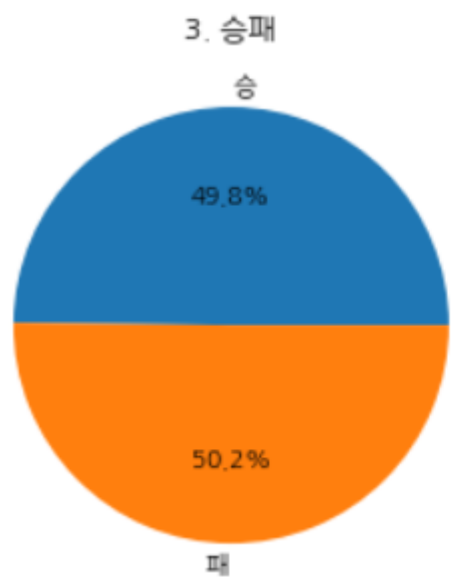


```
# 2. 상대팀 당 경기수
team = []
game_count2 = []
team_game = volleyball['상대팀'].value_counts().to_dict()
for k, v in team_game.items():
    team.append(k)
    game_count2.append(v)
plt.subplot(1, 1, 1)
plt.pie(game_count2, labels=team, autopct='%0.1f%%')
plt.title("2. 상대팀 당 경기수")
plt.show()
```



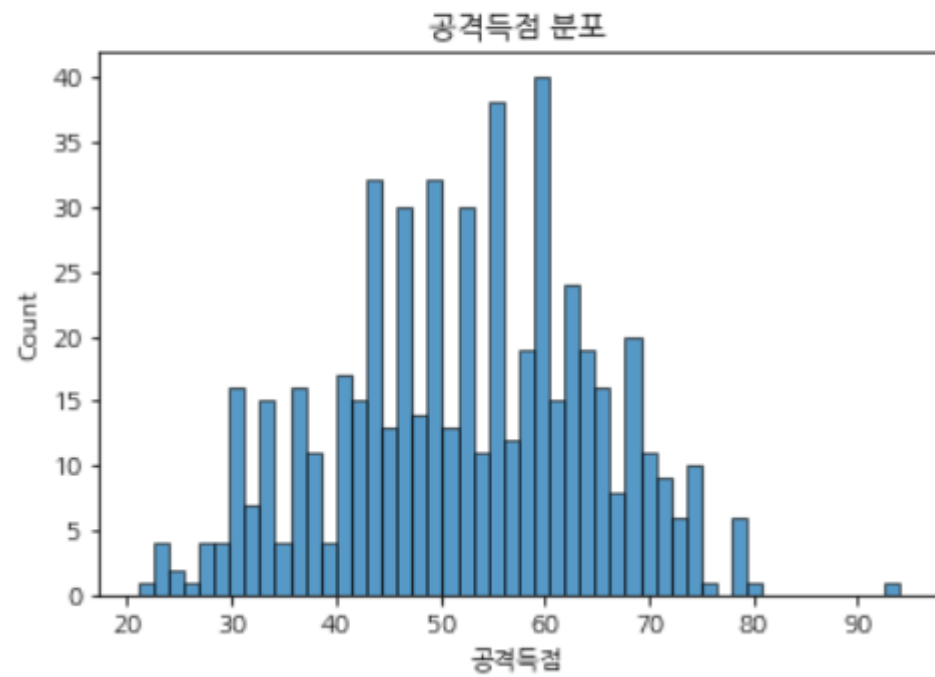
데이터 분석

```
# 3. 승패
win_loss = [len(volleyball[volleyball['결과'] == '승']), len(volleyball[volleyball['결과'] == '패'])]
label = ['승', '패']
plt.subplot(1, 1, 1)
plt.pie(win_loss, labels=label, autopct='%0.1f%%')
plt.title("3. 승패")
plt.show()
```



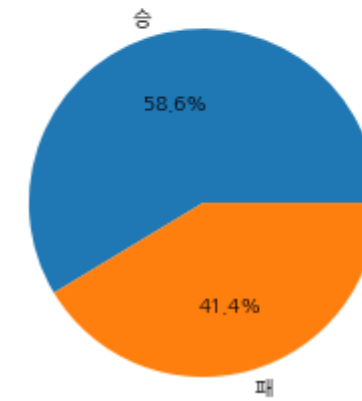
데이터 분석

```
sns.histplot(data=volleyball, x='공격득점', bins=50)
plt.title("공격득점 분포")
plt.show()
```



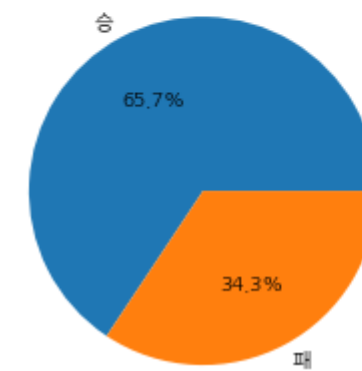
```
attack = volleyball[volleyball['공격득점'] >= 50]
attack_data = [len(attack[attack['결과'] == '승']), len(attack[attack['결과'] == '패'])]
attack_label = ['승', '패']
plt.pie(attack_data, labels=attack_label, autopct='%.1f%')
plt.title("공격득점 50점 이하 결과 비율")
plt.show()
```

공격득점 50점 이하 결과 비율



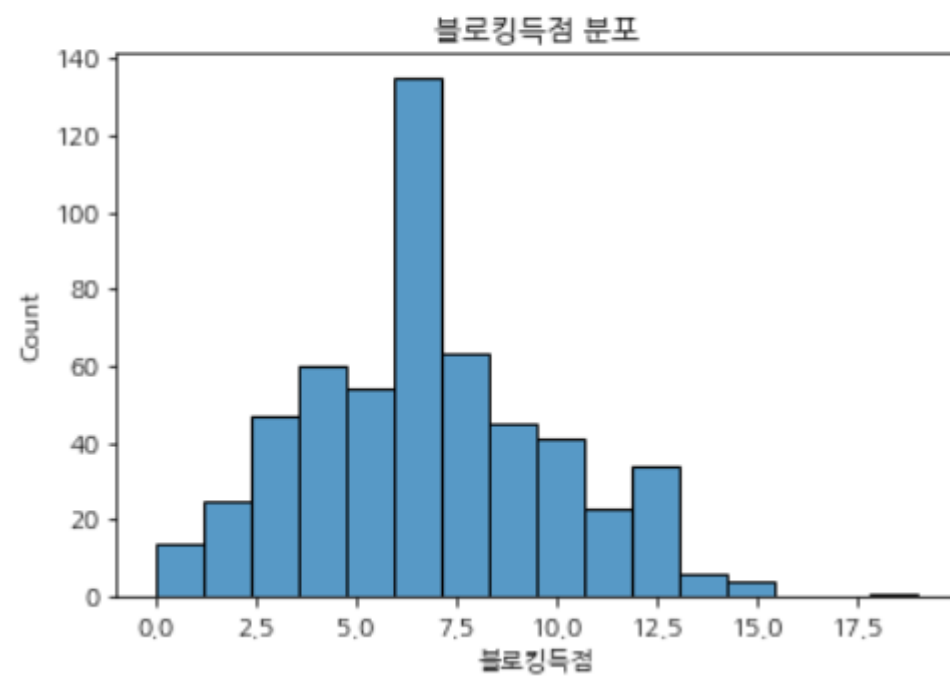
```
attack2 = volleyball[volleyball['공격득점'] >= 60]
attack_data2 = [len(attack2[attack2['결과'] == '승']), len(attack2[attack2['결과'] == '패'])]
attack_label2 = ['승', '패']
plt.pie(attack_data2, labels=attack_label2, autopct='%.1f%')
plt.title("공격득점 60점 이상 결과 비율")
plt.show()
```

공격득점 60점 이상 결과 비율



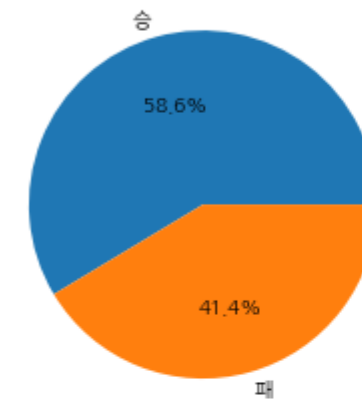
데이터 분석

```
sns.histplot(data=volleyball, x='블로킹득점')  
plt.title("블로킹득점 분포")  
plt.show()
```



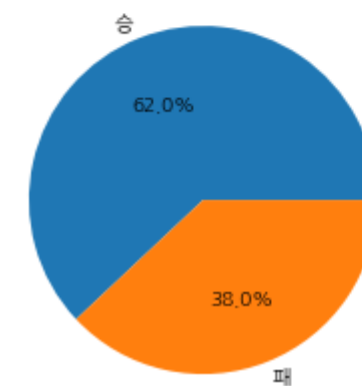
```
block = volleyball[volleyball['블로킹득점'] >= 5]  
block_data = [len(block[block['결과'] == '승']), len(block[block['결과'] == '패'])]  
block_label = ['승', '패']  
plt.pie(block_data, labels=block_label, autopct='%1.1f%%')  
plt.title("블로킹득점 5점 이상 결과 비율")  
plt.show()
```

블로킹득점 5점 이상 결과 비율



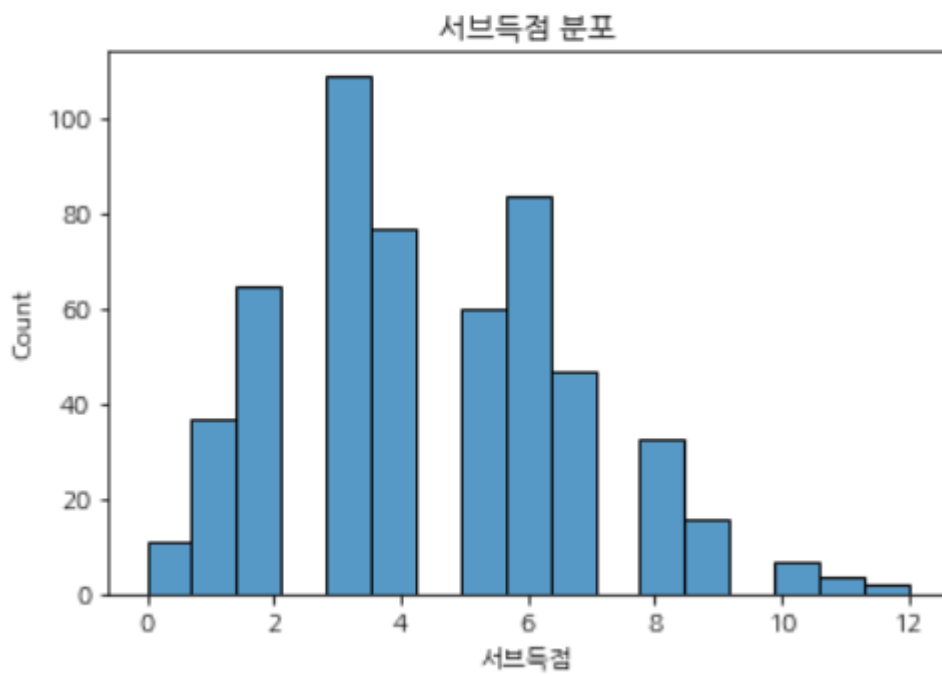
```
block2 = volleyball[volleyball['블로킹득점'] >= 7]  
block_data2 = [len(block2[block2['결과'] == '승']), len(block2[block2['결과'] == '패'])]  
block_label2 = ['승', '패']  
plt.pie(block_data2, labels=block_label2, autopct='%1.1f%%')  
plt.title("블로킹득점 7점 이상 결과 비율")  
plt.show()
```

블로킹득점 7점 이상 결과 비율



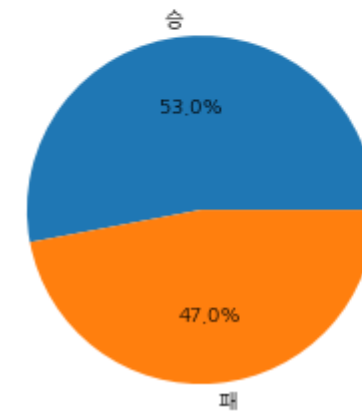
데이터 분석

```
sns.histplot(data=volleyball, x='서브득점')  
plt.title("서브득점 분포")  
plt.show()
```



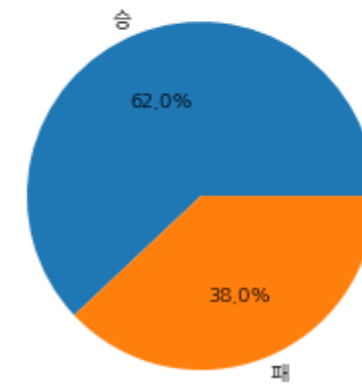
```
serve = volleyball[volleyball['서브득점'] >= 2]  
serve_data = [len(serve[serve['결과'] == '승']), len(serve[serve['결과'] == '패'])]  
serve_label = ['승', '패']  
plt.pie(serve_data, labels=serve_label, autopct='%1f%%')  
plt.title("서브득점 2점 이상 결과 비율")  
plt.show()
```

서브득점 2점 이상 결과 비율



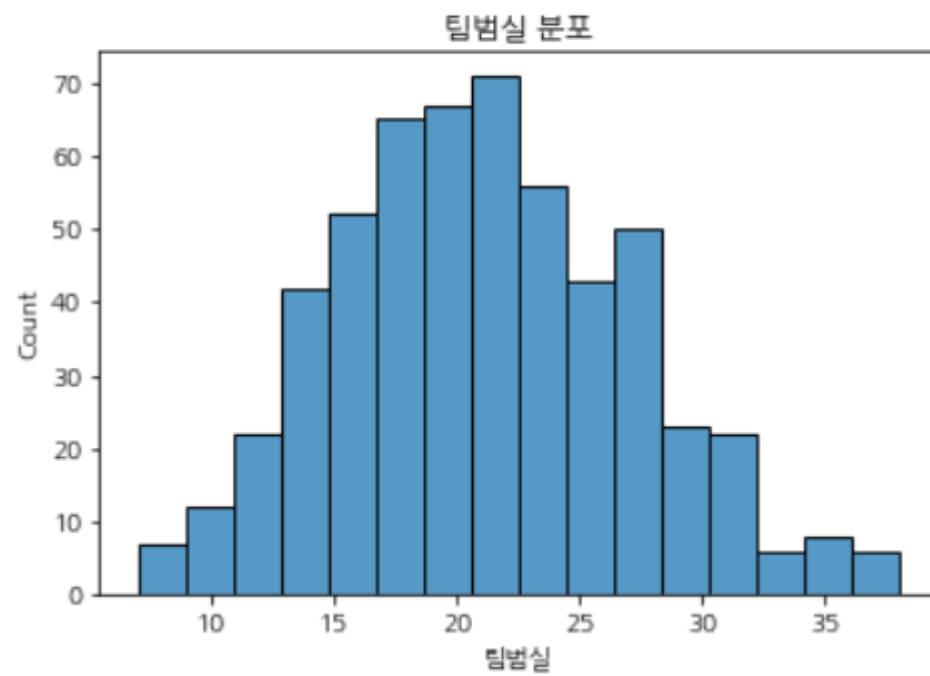
```
serve2 = volleyball[volleyball['블로킹득점'] >= 7]  
serve_data2 = [len(serve2[serve2['결과'] == '승']), len(serve2[serve2['결과'] == '패'])]  
serve_label2 = ['승', '패']  
plt.pie(serve_data2, labels=serve_label2, autopct='%1f%%')  
plt.title("블로킹득점 7점 이상 결과 비율")  
plt.show()
```

블로킹득점 7점 이상 결과 비율



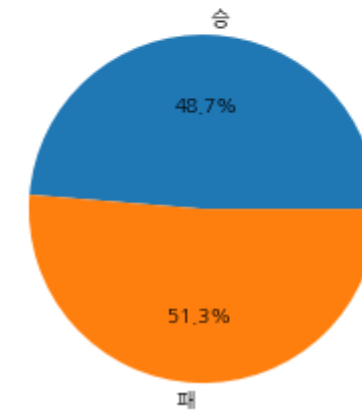
데이터 분석

```
sns.histplot(data=volleyball, x='팀범실')  
plt.title("팀범실 분포")  
plt.show()
```



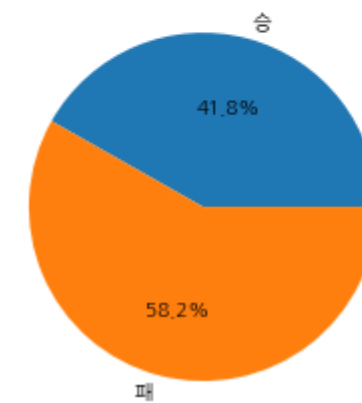
```
miss = volleyball[volleyball['팀범실'] >= 10]  
miss_data = [len(miss[miss['결과'] == '승']), len(miss[miss['결과'] == '패'])]  
miss_label = ['승', '패']  
plt.pie(miss_data, labels=miss_label, autopct='%1f%%')  
plt.title("팀범실 10점 이상 결과 비율")  
plt.show()
```

팀범실 10점 이상 결과 비율



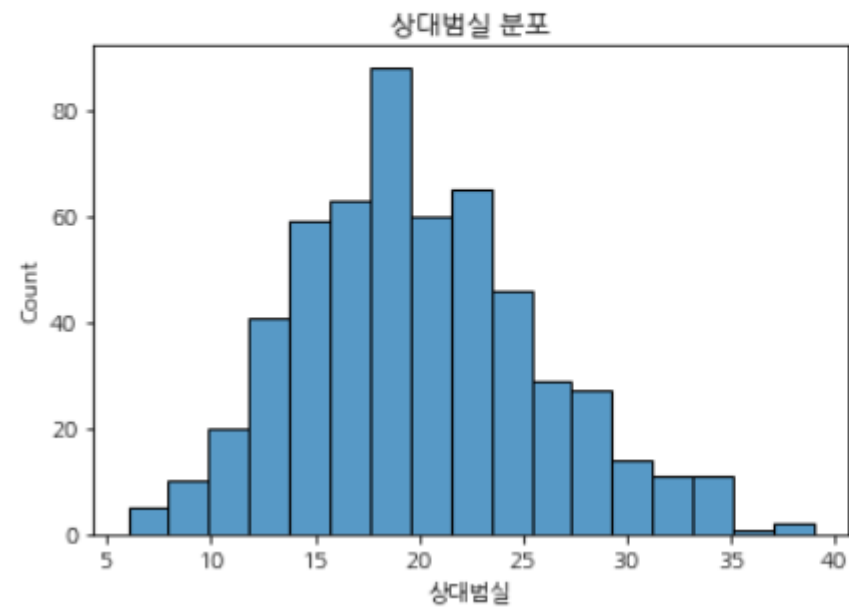
```
miss2 = volleyball[volleyball['팀범실'] >= 20]  
miss_data2 = [len(miss2[miss2['결과'] == '승']), len(miss2[miss2['결과'] == '패'])]  
miss_label2 = ['승', '패']  
plt.pie(miss_data2, labels=miss_label2, autopct='%1f%%')  
plt.title("팀범실 20점 이상 결과 비율")  
plt.show()
```

팀범실 20점 이상 결과 비율

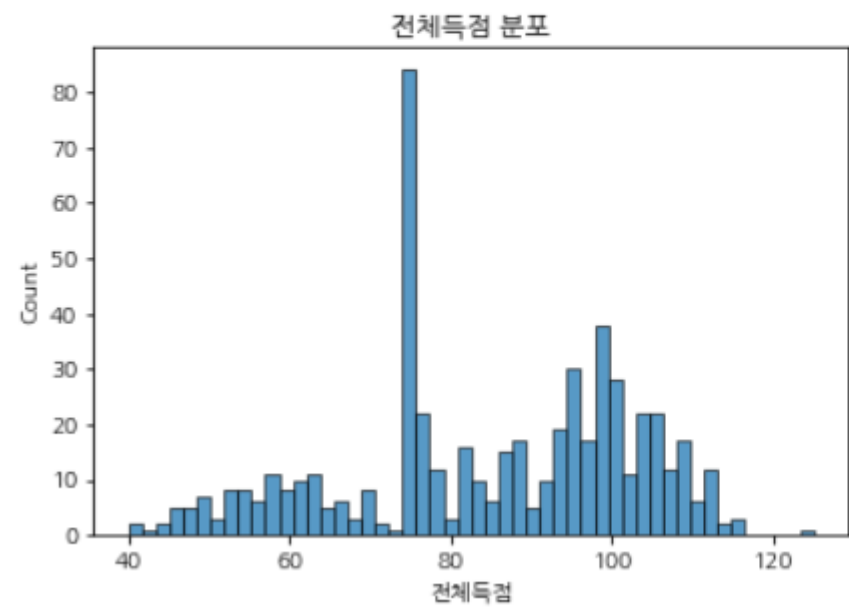


데이터 분석

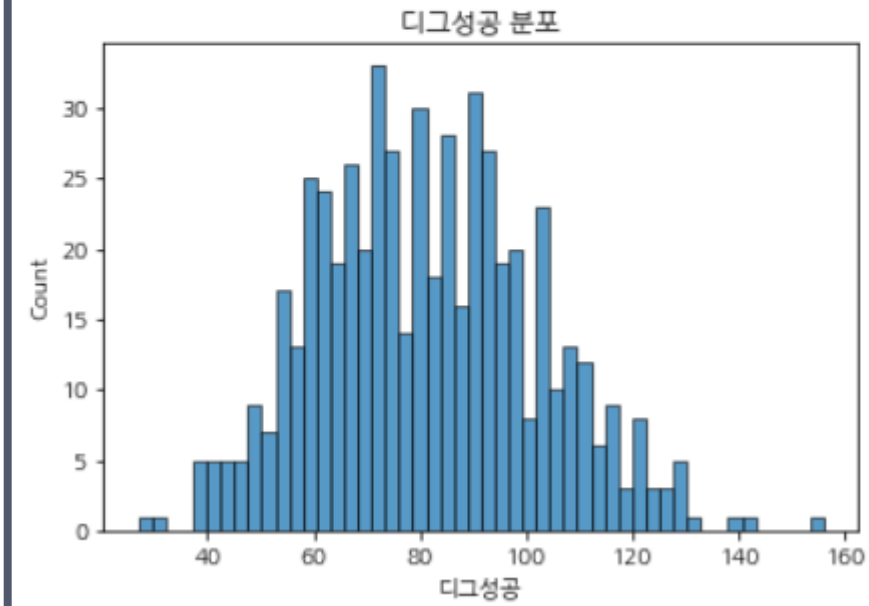
```
sns.histplot(data=volleyball, x='상대범실')  
plt.title("상대범실 분포")  
plt.show()
```



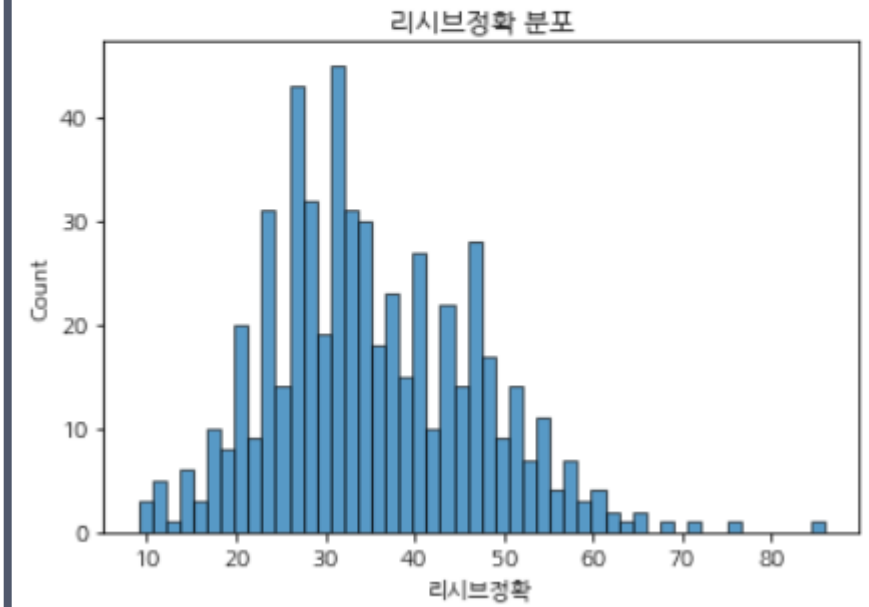
```
sns.histplot(data=volleyball, x='전체득점', bins=50)  
plt.title("전체득점 분포")  
plt.show()
```



```
sns.histplot(data=volleyball, x='디그성공', bins=50)  
plt.title("디그성공 분포")  
plt.show()
```

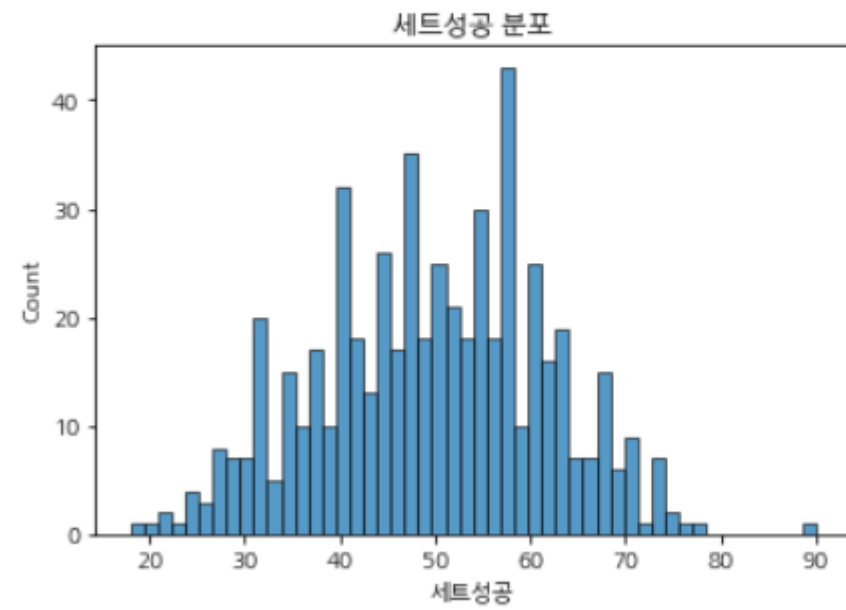


```
sns.histplot(data=volleyball, x='리시브정확', bins=50)  
plt.title("리시브정확 분포")  
plt.show()
```

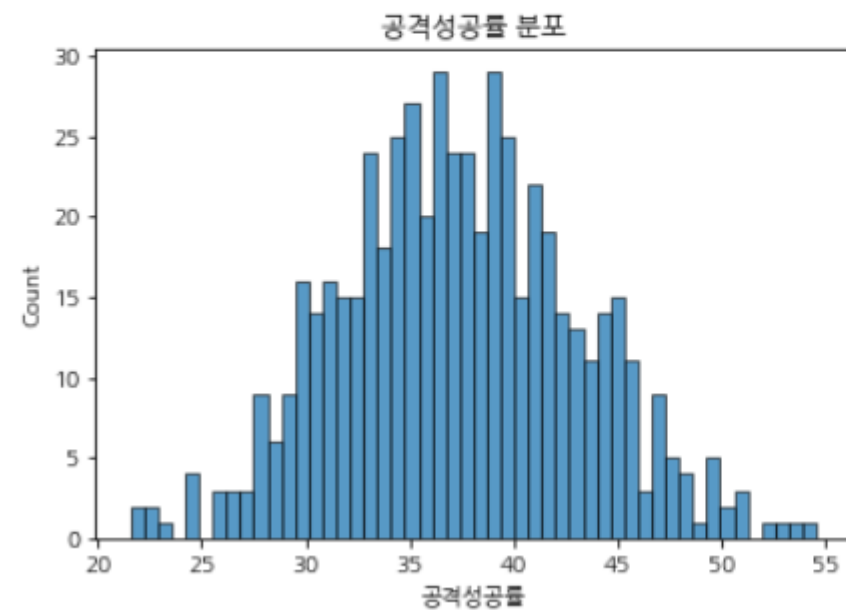


데이터 분석

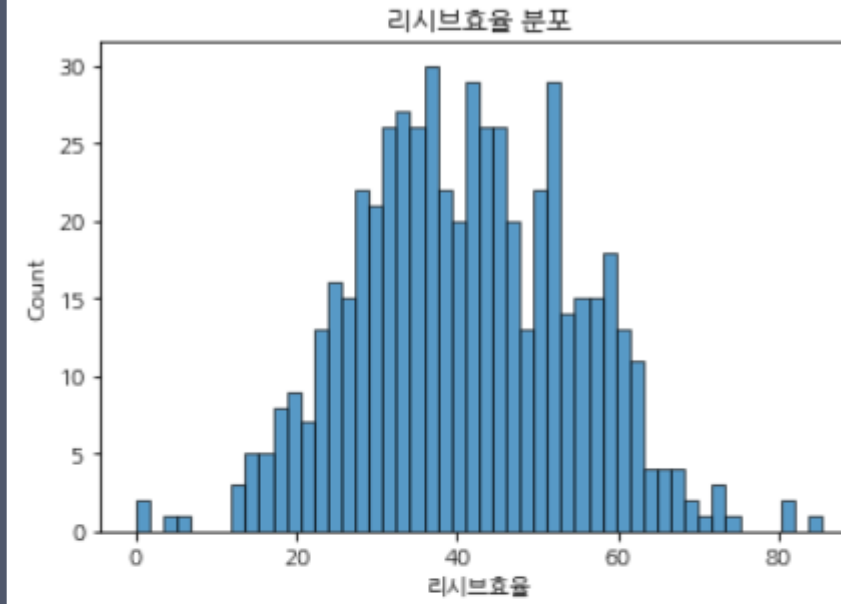
```
sns.histplot(data=volleyball, x='세트성공', bins=50)  
plt.title("세트성공 분포")  
plt.show()
```



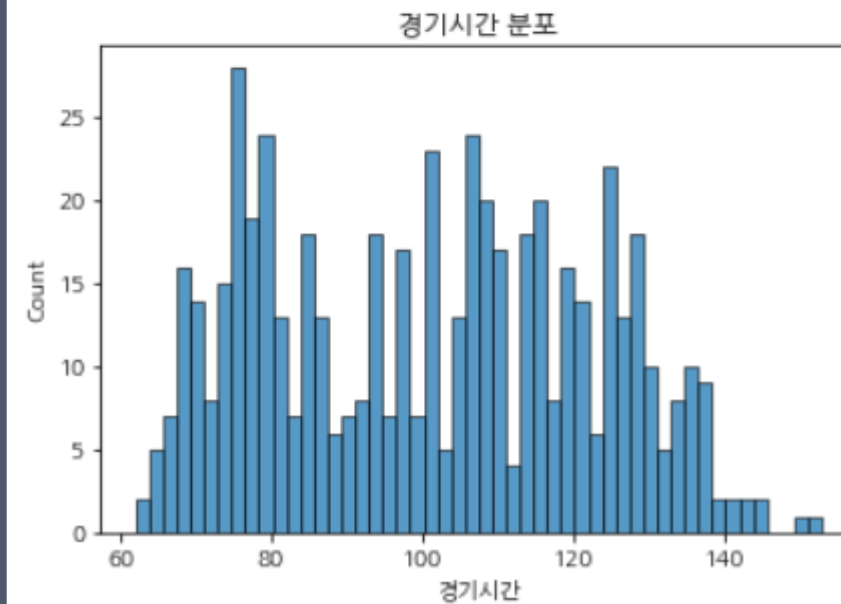
```
sns.histplot(data=volleyball, x='공격성공률', bins=50)  
plt.title("공격성공률 분포")  
plt.show()
```



```
sns.histplot(data=volleyball, x='리시브효율', bins=50)  
plt.title("리시브효율 분포")  
plt.show()
```



```
sns.histplot(data=volleyball, x='경기시간', bins=50)  
plt.title("경기시간 분포")  
plt.show()
```



데이터 분석

```
df1 = volleyball[['공격득점', '블로킹득점', '서브득점', '상대범실']]
df1.plot(kind='barh', stacked=True)
plt.title("득점 비율")
plt.xlabel("득점")
plt.show()
```

