Kaggle Competition

Kobe Bryant Shot Selection

Department of SNS

HAN JAEYOON

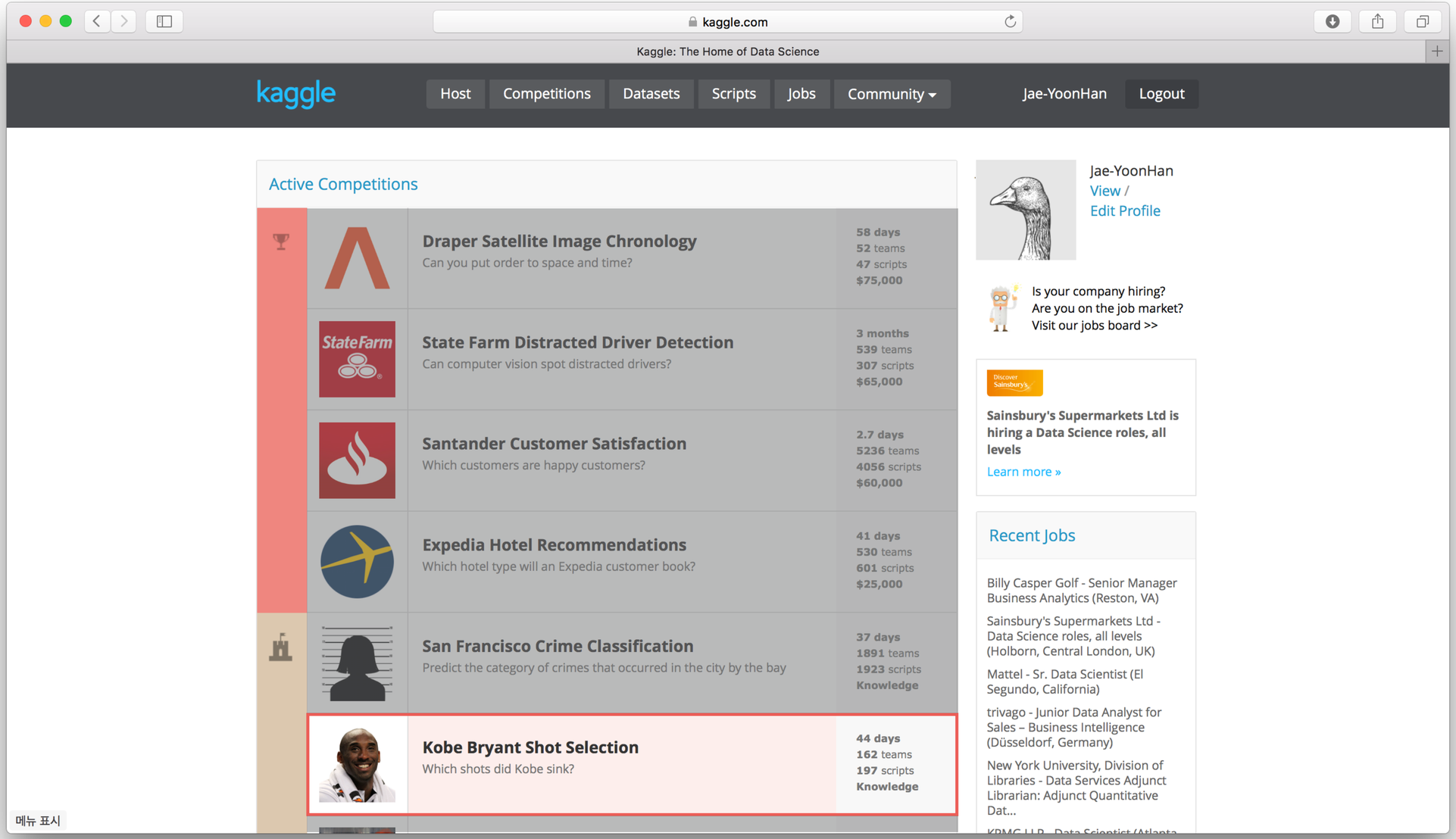
CONTENTS

01 서론

kaggle™

The Home of Data Science

전세계 데이터 과학자들이 특정 문제의 해결법을 두고 경쟁을 벌이는 온라인 플랫폼


kaggle.com
Kaggle: The Home of Data Science
kaggle
Host
Competitions
Datasets
Scripts
Jobs
Community
Jae-YoonHan
Logout
Active Competitions
Draper Satellite Image Chronology
Can you put order to space and time?
58 days
52 teams
47 scripts
$75,000
State Farm Distracted Driver Detection
Can computer vision spot distracted drivers?
3 months
539 teams
307 scripts
$65,000
Santander Customer Satisfaction
Which customers are happy customers?
2.7 days
5236 teams
4056 scripts
$60,000
Expedia Hotel Recommendations
Which hotel type will an Expedia customer book?
41 days
530 teams
601 scripts
$25,000
San Francisco Crime Classification
Predict the category of crimes that occurred in the city by the bay
37 days
1891 teams
1923 scripts
Knowledge
Kobe Bryant Shot Selection
Which shots did Kobe sink?
44 days
162 teams
197 scripts
Knowledge
Jae-YoonHan
View /
Edit Profile
Is your company hiring?
Are you on the job market?
Visit our jobs board >>
Sainsbury's Supermarkets Ltd is hiring a Data Science roles, all levels
Learn more »
Recent Jobs
Billy Casper Golf - Senior Manager Business Analytics (Reston, VA)
Sainsbury's Supermarkets Ltd - Data Science roles, all levels (Holborn, Central London, UK)
Mattel - Sr. Data Scientist (El Segundo, California)
trivago - Junior Data Analyst for Sales – Business Intelligence (Düsseldorf, Germany)
New York University, Division of Libraries - Data Services Adjunct Librarian: Adjunct Quantitative Dat...
메뉴 표시

Kobe Bryant

24

Born : *1978. 08. 23 (Age : 37)*

Debut : *1996. 11. 03*

Olympic Gold Medal : *2 Times*

5 Times NBA Championship

2 Times NBA Final MVP

All NBA First Team : *11 Times*

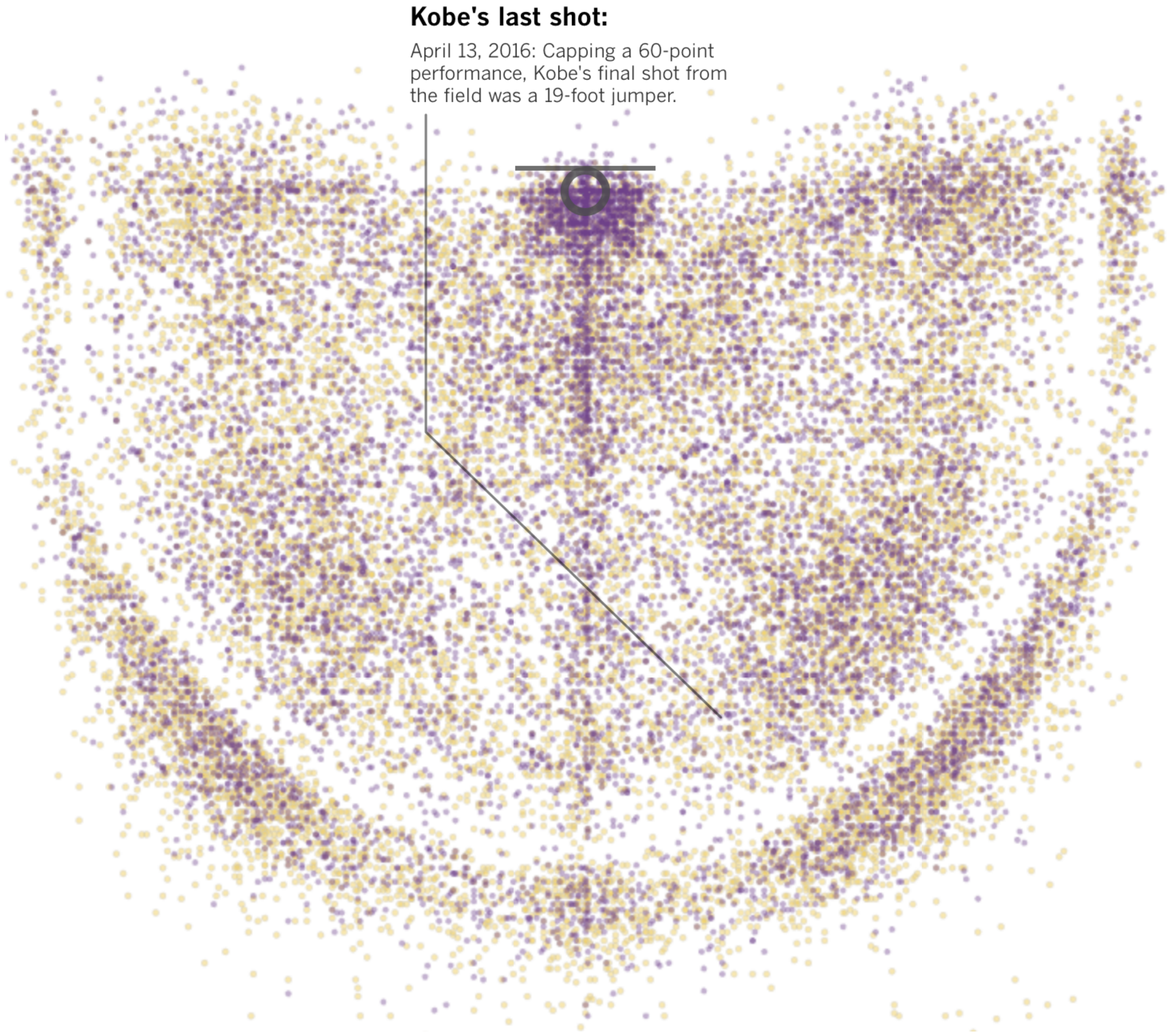
NBA All-Star : *18 Times*

Total Score : *33,643*

FG% : *44.7%* / PTS : *25.0*

Bryant attempted 30,699 shots throughout his career.

Made
Missed



http://graphics.latimes.com/kobe-every-shot-ever/


STAPLES Center
Coca-Cola
Coca-Cola
Enter the Sky
STAPLES Center
STAPLES Center

02 데이터 탐색/전처리

R + RStudio

POWERFUL IDE

02 데이터 탐색/전처리

Data Files

File Name	Available Formats
data.csv	.zip (679.31 kb)
sample_submission.csv	.zip (12.38 kb)

30,697개의 슛 데이터
- **25개**의 어트리뷰트(Attribute)

This data contains the location and circumstances of every field goal attempted by Kobe Bryant took during his 20-year career. Your task is to predict whether the basket went in (shot_made_flag).

We have removed 5000 of the shot_made_flags (represented as missing values in the csv file). These are the test set shots for which you must submit a prediction. You are provided a sample submission file with the correct shot_ids needed for a valid prediction.

Train Data : 25,679개의 슛 데이터
Test Data : 5,000개의 슛 데이터

To avoid leakage, your method should only train on events that occurred prior to the shot for which you are predicting! Since this is a playground competition with public answers, it's up to you to abide by this rule.

Data Leakage?

Data Leakage

https://www.kaggle.com/wiki/Leakage

변수명	설명
action_type	시도한 슛의 세부 종류
combined_shot_type	시도한 슛의 종류
game_event_id	해당 경기에서 발생한 이벤트 중 시도한 슛의 순번
game_id	경기 식별자
lat	슛을 시도한 위치의 위도
loc_x	경기장 x 좌표
loc_y	경기장 y 좌표
lon	슛을 시도한 위치의 경도
minutes_remaining	쿼터의 남은 시간 중 분(Minutes)
period	쿼터 (1~4)
playoffs	플레이오프 경기 여부
season	해당 시즌

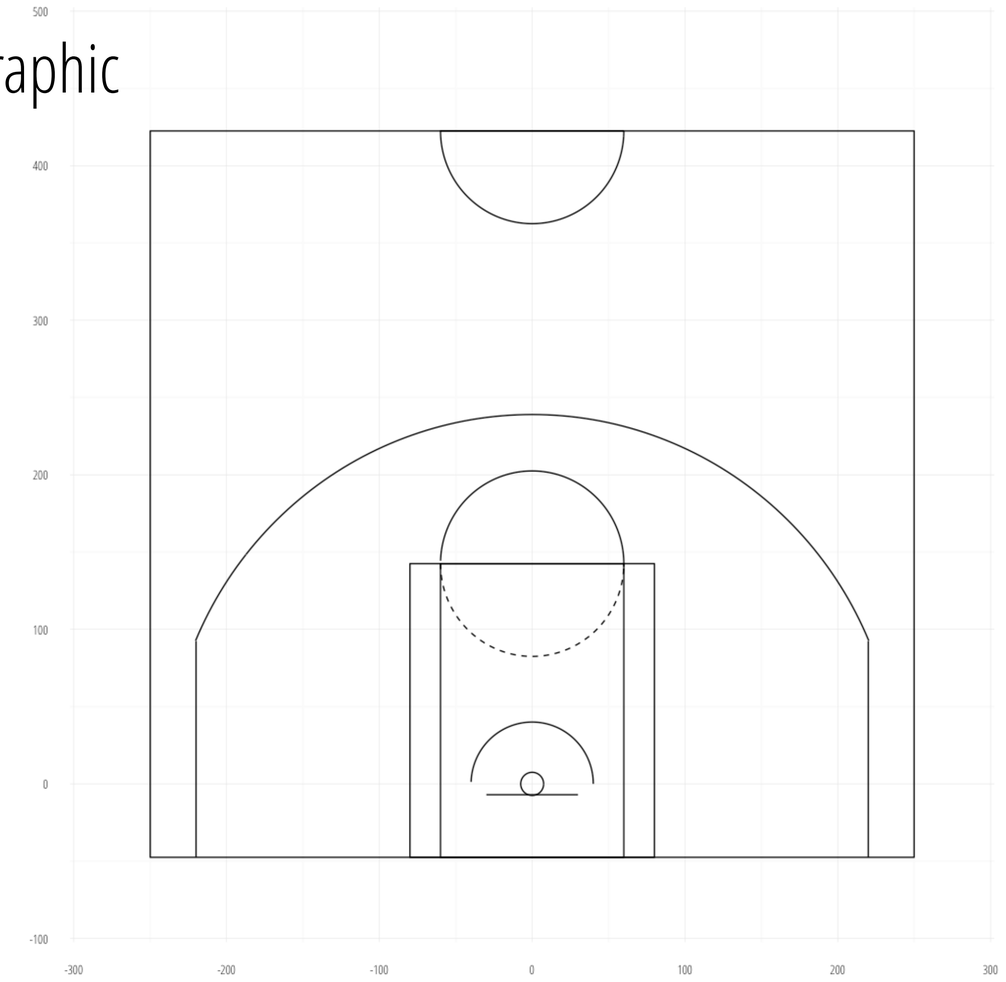
변수명	설명
seconds_remaining	쿼터의 남은 시간 중 초(Seconds)
shot_distance	슛을 시도한 위치로부터 림(Rim)까지의 거리
shot_made_flag	슛 성공 여부
shot_type	시도한 슛의 점수 (2점, 3점)
shot_zone_area	슛을 시도한 구역 (좌우 기준)
shot_zone_basic	슛을 시도한 구역의 명칭
shot_zone_range	슛을 시도한 위치로부터 림(Rim)까지의 거리, 범주형 데이터
team_id	팀 식별자
team_name	팀 이름
game_date	경기 일자
matchup	대진
opponent	상대팀 이름 (약자)
shot_id	슛 식별자

02 데이터 탐색/전처리

ggplot2

The Grammar of Graphic

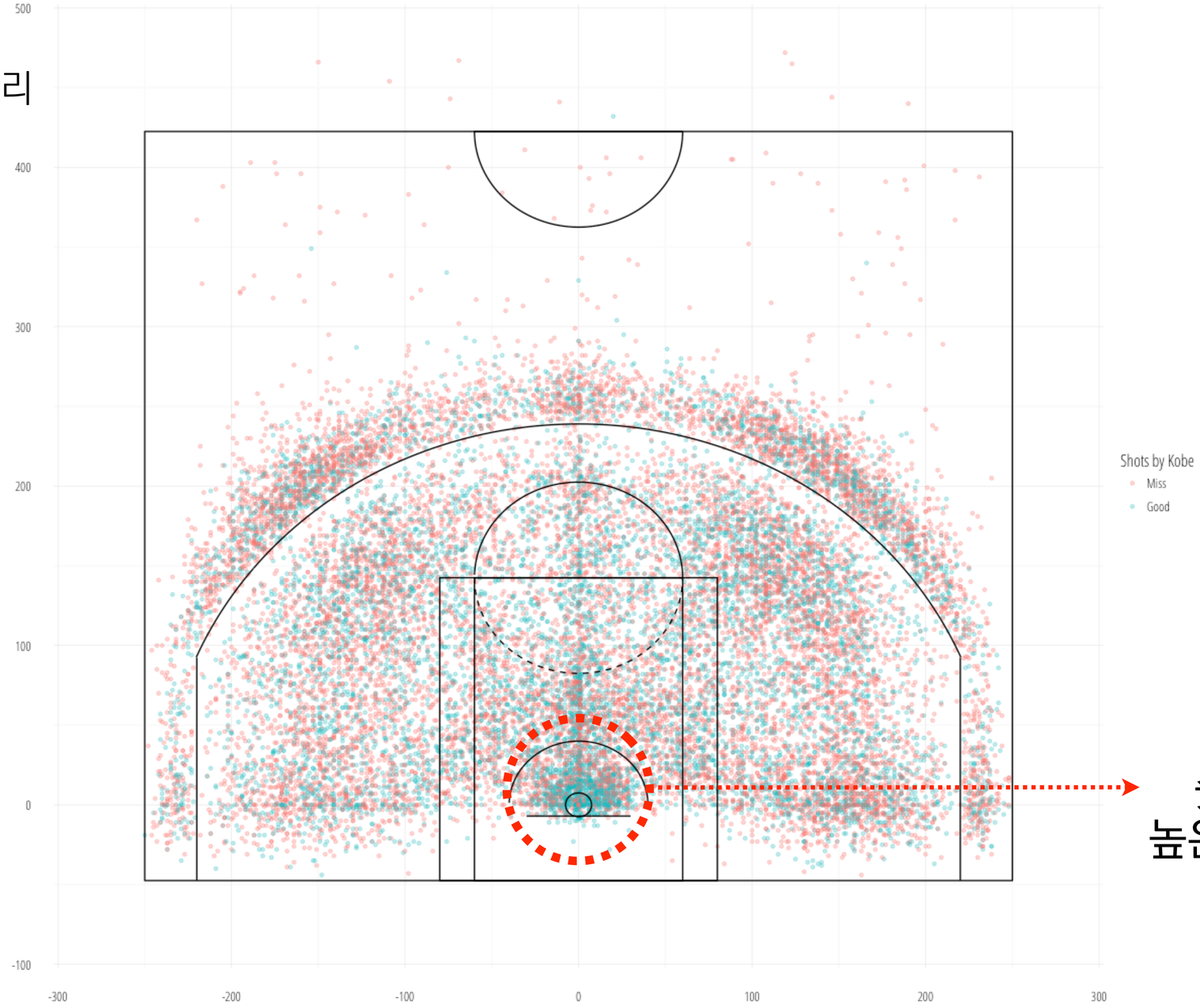
```r
library(dplyr)
library(ggplot2)
library(ggthemes)
library(packcircles)

# Make a dataset for drawing hoop and backboard
circle <- as.data.frame(circleVertices(0, 0, 7.5, npoints = 100))
backboard <- data.frame(x = c(-30, 30), y = c(-7, -7))

# Make a dataset for drawing free throw top arc & bottom arc
circle.FT <- as.data.frame(circleVertices(0, 142.5, 60, npoint = 200))
circle.FT.top <- circle.FT %>%
        filter(y >= 142.5)
circle.FT.bottom <- circle.FT %>%
        filter(y < 142.5)

# Make a dataset for drawing restricted zone
restricted <- as.data.frame(circleVertices(0, 0, 40, npoint = 200))
restricted.top <- restricted %>%
        filter(y >= 0)

# Make a dataset for drawing three point line
left_corner_three <- data.frame(x = c(-220, -220), y = c(-47.5, 92.5))
right_corner_three <- data.frame(x = c(220, 220), y = c(-47.5, 92.5))
three_point <- as.data.frame(circleVertices(0, 0, 239, npoint = 300))
three_point <- three_point %>%
        filter(y >= 88)

# Make a dataset for drawing center court arc
center_arc <- as.data.frame(circleVertices(0, 422.5, 60, npoint = 200))
center_arc <- center_arc %>%
        filter(y <= 422.5)

basketball_court <- ggplot() +
        geom_rect(mapping = aes(xmin = -250, xmax = 250, ymin = -47.5, ymax = 422.5), colour = "black", alpha = 0) + # outer line
        geom_path(data = circle, aes(x = x, y = y)) + # Hoop
        geom_path(data = backboard, aes(x = x, y = y)) + # Backboard
        geom_rect(mapping = aes(xmin = -80, xmax = 80, ymin = -47.5, ymax = 142.5), colour = "black", alpha = 0) + # outer_paint_zone
        geom_rect(mapping = aes(xmin = -60, xmax = 60, ymin = -47.5, ymax = 142.5), colour = "black", alpha = 0) + # inner_paint_zone
        geom_path(data = circle.FT.top, aes(x = x, y = y)) +  # FT top arc
        geom_path(data = circle.FT.bottom, aes(x = x, y = y), linetype = 2) + # FT bottom arc
        geom_path(data = restricted.top, aes(x = x, y = y)) +  # restricted zone
        geom_path(data = left_corner_three, aes(x = x, y = y)) + # left_corner_three
        geom_path(data = right_corner_three, aes(x = x, y = y)) + # right_corner_three
        geom_path(data = three_point, aes(x = x, y = y)) + # three_point_line
        geom_path(data = center_arc, aes(x = x, y = y)) + # center_arc
        xlim(-275, 275) + ylim(-75, 475)
basketball_court
```



VISUALIZING the train set

```
kobe_train <- kobe_shot %>%
        filter(!is.na(shot_made_flag))
kobe_test <- kobe_shot %>%
        filter(is.na(shot_made_flag))

ggplot() +
        geom_point(data = kobe_train,
                   aes(x = loc_x, y = loc_y, colour = factor(shot_made_flag),
                       fill = NULL), alpha = 0.3, size = 1) +
        basketball_court$layers +
        xlim(-275, 275) + ylim(-75, 475) +
        scale_color_hue("Shots by Kobe", labels = c("Miss", "Good"))
```

02 데이터 탐색/전처리



골밑에서의
슛 성공률이
높은 것으로 추측

시도한 숫이
너무 **많아서**
유의미한 인사이트를
얻기 힘들다.

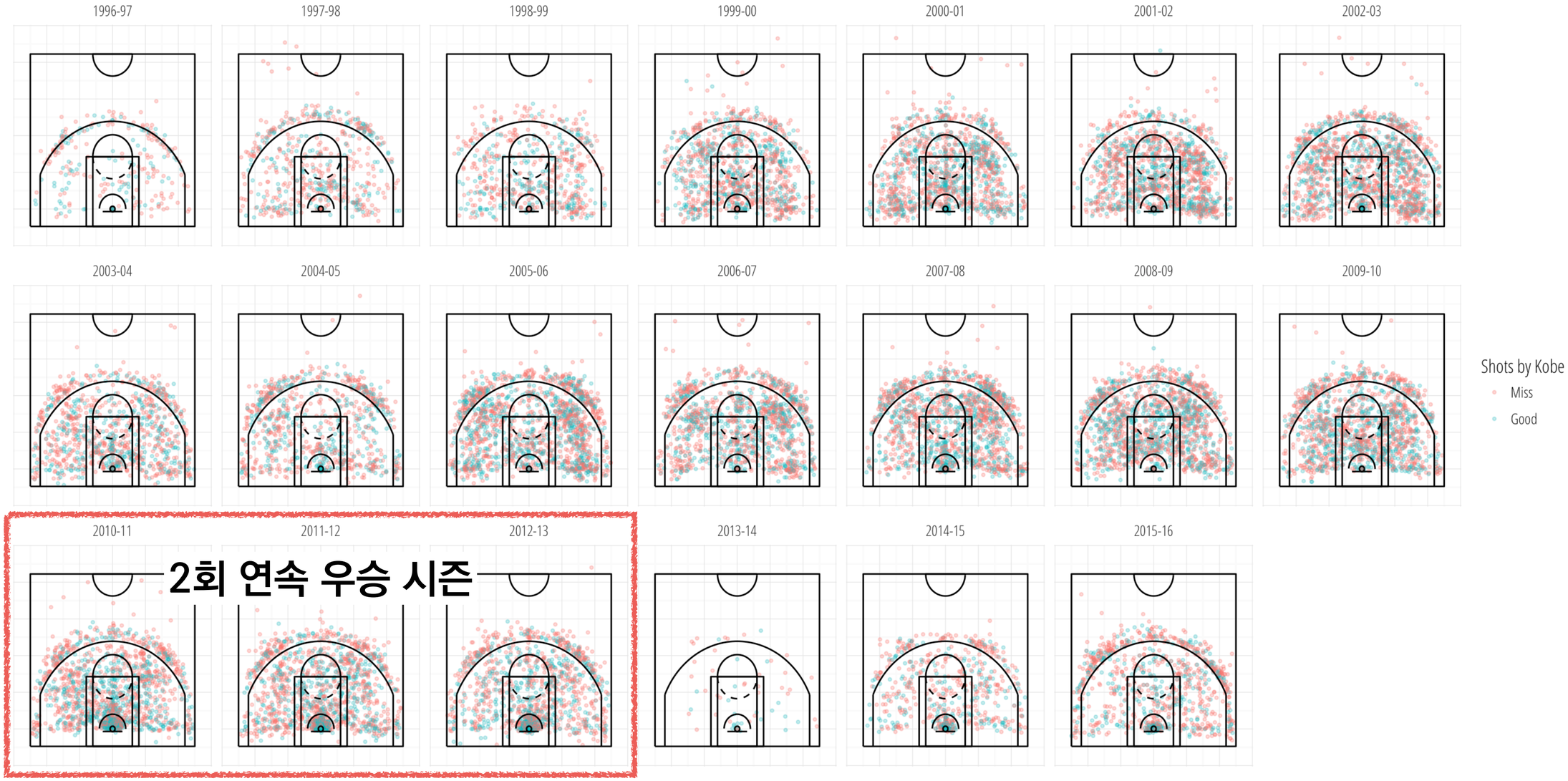
↓

케이스 분류

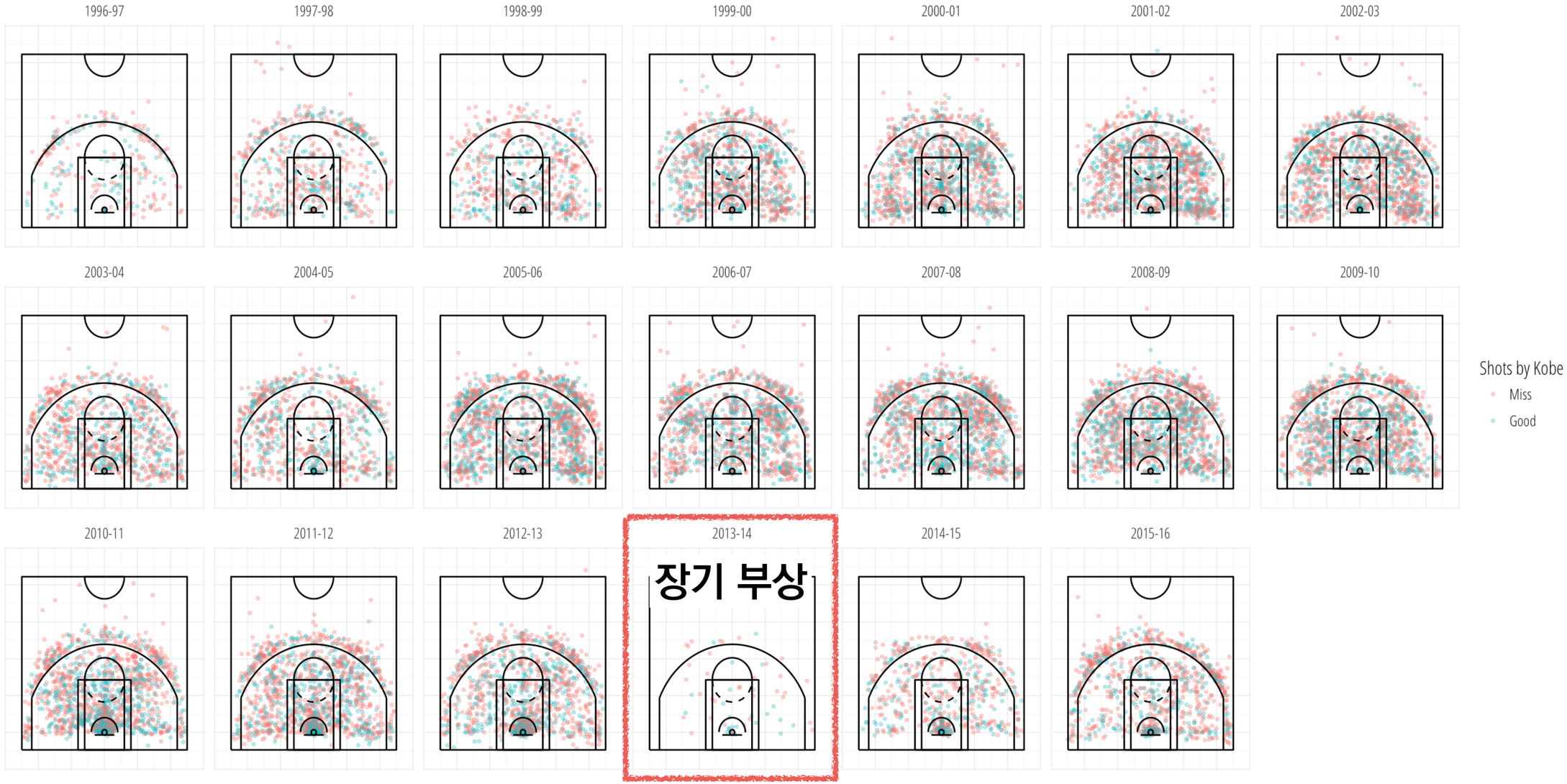
시즌 단위의 데이터 시각화

```
ggplot() +
        xlim(-275, 275) + ylim(-75, 475) +
        geom_point(data = kobe_train,
                   aes(x = loc_x, y = loc_y, colour = factor(shot_made_flag),
                       fill = NULL), alpha = 0.3, size = 0.3) +
        scale_color_hue("Shots by Kobe", labels = c("Miss", "Good")) +
        xlab(NULL) + ylab(NULL) +
        basketball_court$layers +
        facet_wrap( ~ season, nrow = 3)
```

02 데이터 탐색/전처리


1996-97
1997-98
1998-99
1999-00
2000-01
2001-02
2002-03
2003-04
2004-05
2005-06
2006-07
2007-08
2008-09
2009-10
2010-11
2011-12
2012-13
2013-14
2014-15
2015-16
2회 연속 우승 시즌
Shots by Kobe
Miss
Good

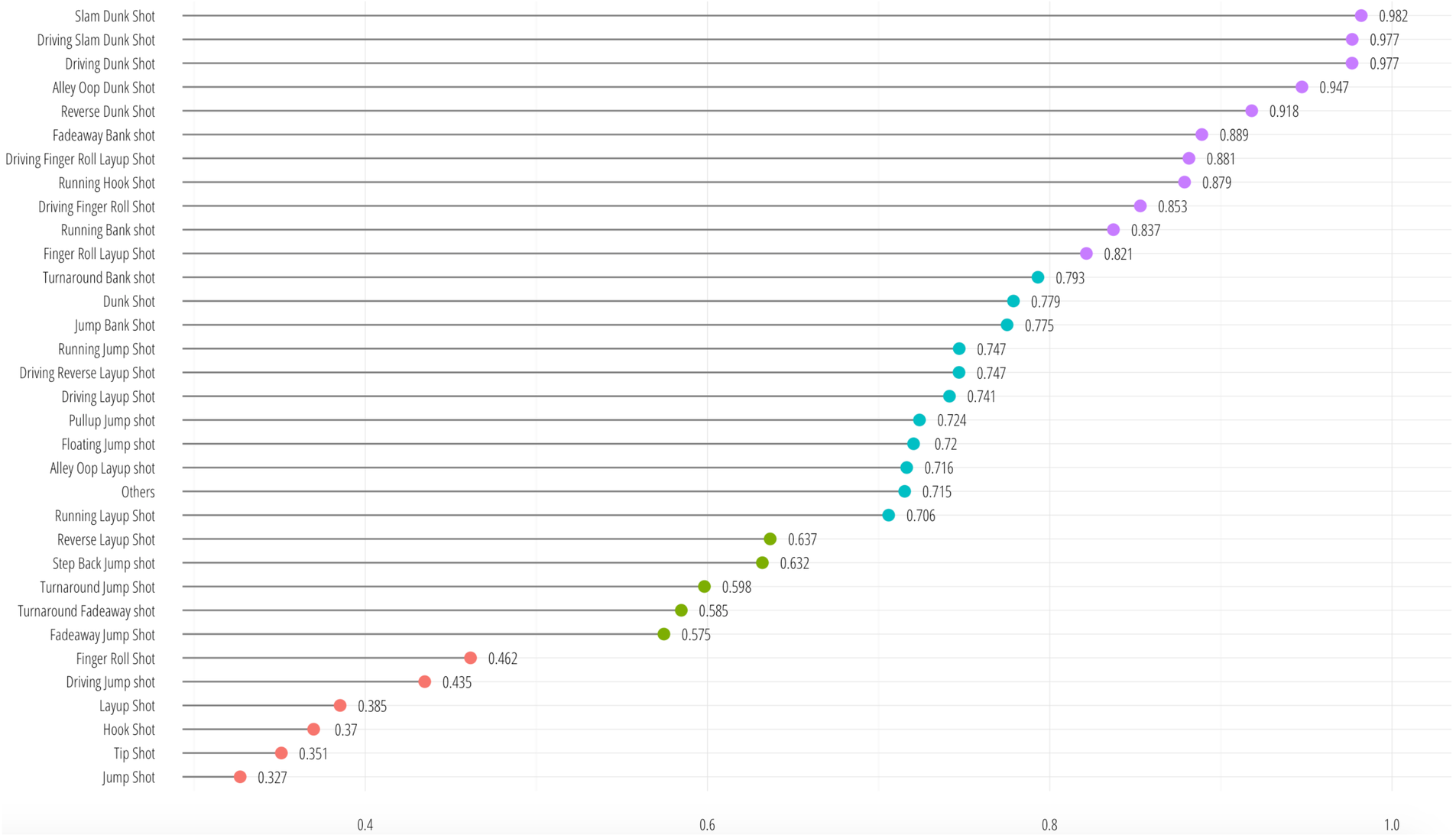
02 데이터 탐색/전처리


1996-97
1997-98
1998-99
1999-00
2000-01
2001-02
2002-03
2003-04
2004-05
2005-06
2006-07
2007-08
2008-09
2009-10
2010-11
2011-12
2012-13
2013-14
장기 부상
2014-15
2015-16
Shots by Kobe
Miss
Good

슛 종류별 시각화

```
shot_type <- kobe_train
less_than_20 <- which(table(shot_type$action_type) < 20)
levels(shot_type$action_type)[less_than_20] <- "Others"
table(shot_type$action_type)

rate_based_shot_type <- shot_type %>%
        group_by(action_type) %>%
        summarise(shot_rate = sum(shot_made_flag) / n()) %>%
        arrange(desc(shot_rate)) %>%
        mutate(color = cut(rate_based_shot_type$shot_rate, 4))

ggplot(data = rate_based_shot_type, aes(x = shot_rate, y = reorder(action_type, shot_rate))) +
        geom_segment(aes(yend = action_type), xend = 0, colour = "grey50") +
        geom_point(size = 3, aes(colour = color)) +
        geom_text(aes(x = shot_rate + 0.019, label = round(shot_rate, 3)), size = 3.5,
                  family="OpenSans-CondensedLight") +
        theme(legend.position="none")
```

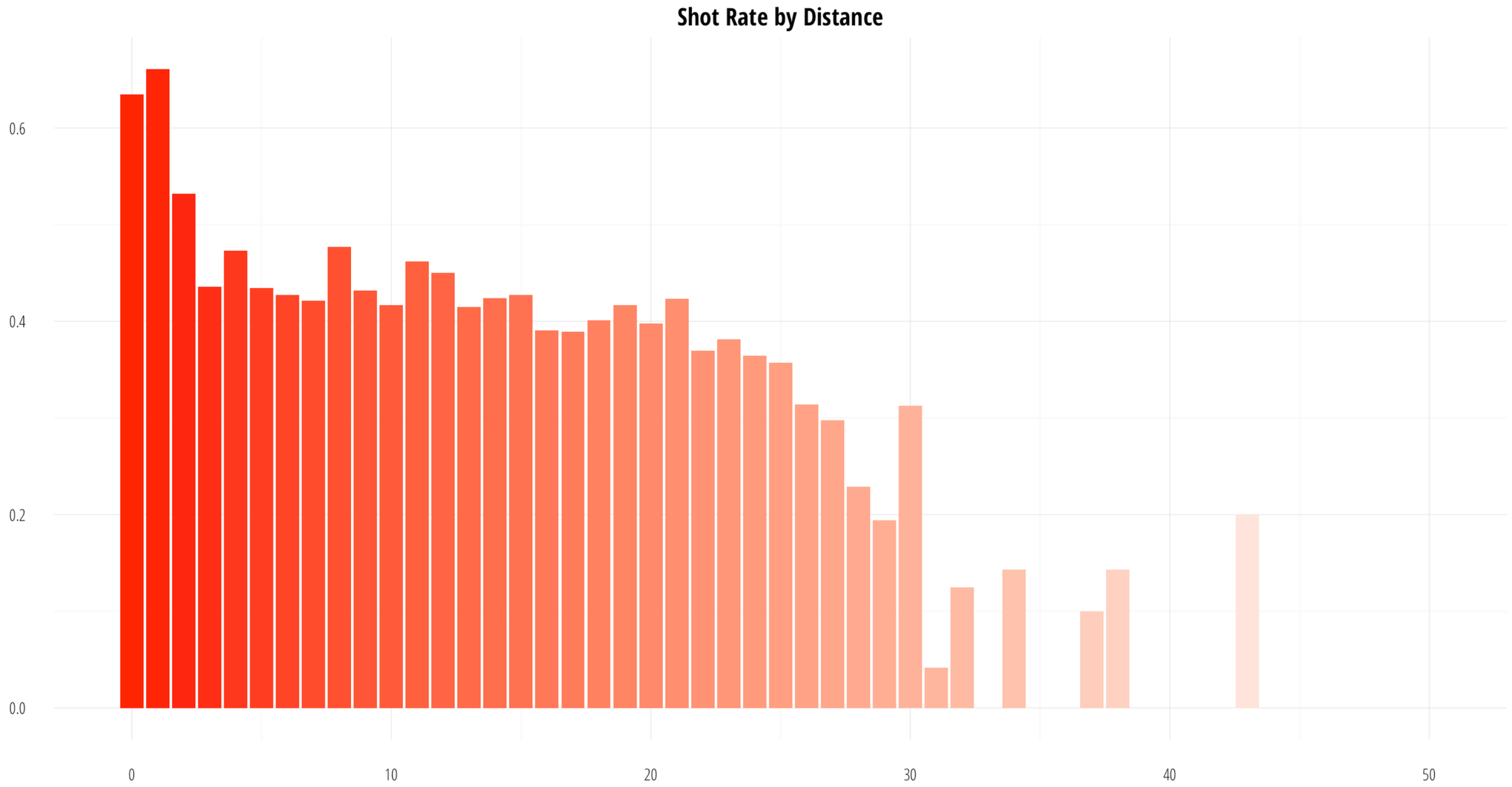
02 데이터 탐색/전처리


Slam Dunk Shot 0.982
Driving Slam Dunk Shot 0.977
Driving Dunk Shot 0.977
Alley Oop Dunk Shot 0.947
Reverse Dunk Shot 0.918
Fadeaway Bank shot 0.889
Driving Finger Roll Layup Shot 0.881
Running Hook Shot 0.879
Driving Finger Roll Shot 0.853
Running Bank shot 0.837
Finger Roll Layup Shot 0.821
Turnaround Bank shot 0.793
Dunk Shot 0.779
Jump Bank Shot 0.775
Running Jump Shot 0.747
Driving Reverse Layup Shot 0.747
Driving Layup Shot 0.741
Pullup Jump shot 0.724
Floating Jump shot 0.72
Alley Oop Layup shot 0.716
Others 0.715
Running Layup Shot 0.706
Reverse Layup Shot 0.637
Step Back Jump shot 0.632
Turnaround Jump Shot 0.598
Turnaround Fadeaway shot 0.585
Fadeaway Jump Shot 0.575
Finger Roll Shot 0.462
Driving Jump shot 0.435
Layup Shot 0.385
Hook Shot 0.37
Tip Shot 0.351
Jump Shot 0.327
0.4
0.6
0.8
1.0

슛 거리별 시각화

```
distance_based <- kobe_train %>%
        group_by(shot_distance) %>%
        summarise(rate = sum(shot_made_flag) / n()) %>%
        filter(shot_distance <= 50)

ggplot(data = distance_based, aes(x = shot_distance, y = rate, fill = shot_distance)) +
        geom_bar(stat = "identity") +
        scale_fill_gradient(low="red", high="white") +
        theme(legend.position="none") +
        ggtitle("Shot Rate by Distance")
```

02 데이터 탐색/전처리



선수들의 **체력적인 요소**를 고려해야 하지 않을까?

휴식을 취한 기간이 길었다면
슛 정확도가 **올라가지** 않을까?

휴식 기간이 너무 길었다면
경기 감각이 **떨어지지** 않을까?

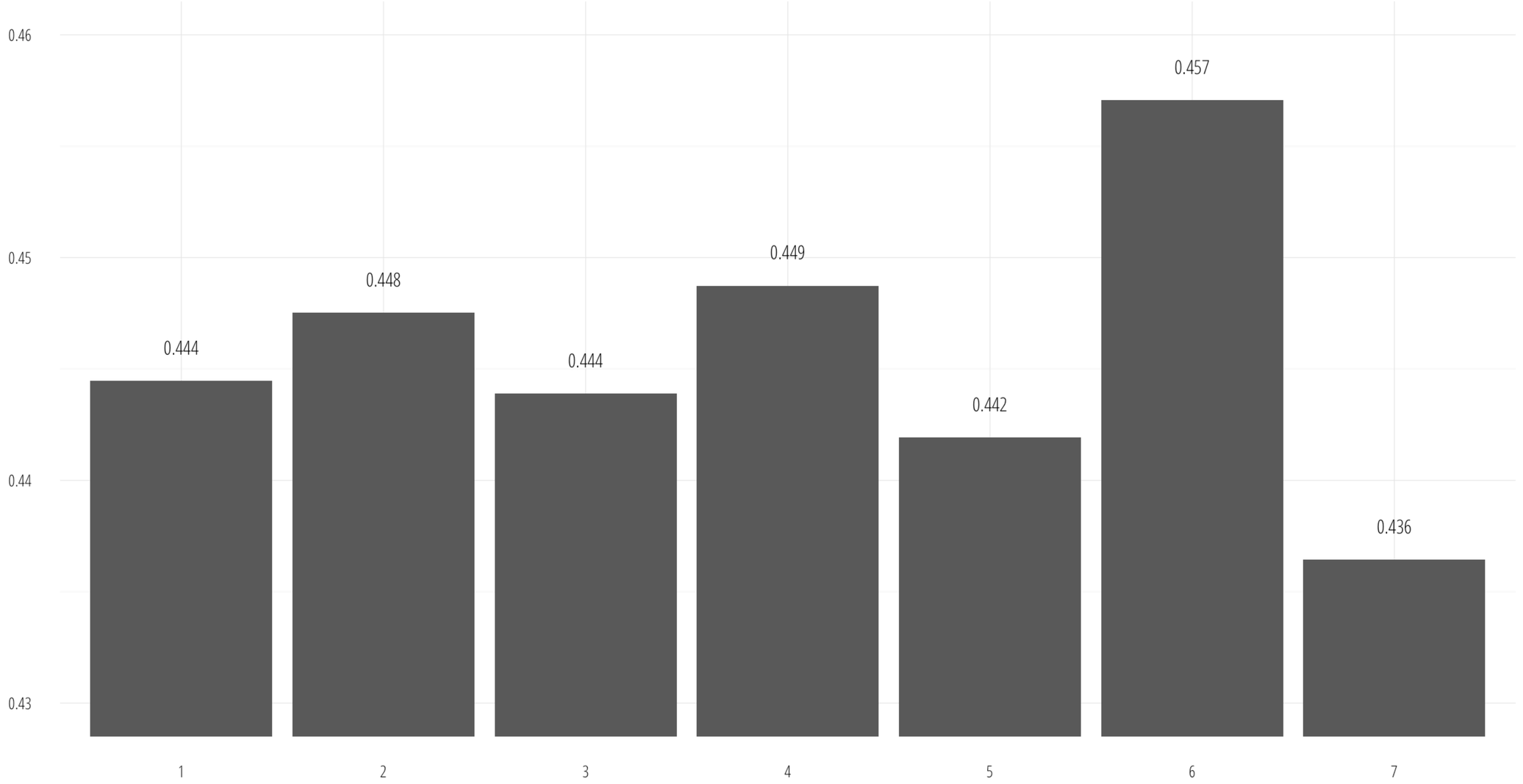
```r
library(lubridate)
kobe_shot$game_date <- as.Date(kobe_shot$game_date)
unique_date <- unique(kobe_shot$game_date)
day_difference <- c(0, diff(unique_date))
kobe_shot <- kobe_shot %>%
        mutate(day_difference = factor(kobe_shot$game_id))

levels(kobe_shot$day_difference) <- day_difference
kobe_shot$day_difference <- as.numeric(as.character(kobe_shot$day_difference))
table(kobe_shot$day_difference)
```

```
-7101 -5505     0     1     2     3     4     5     6     7     8     9    10    11    12    14    18
    1     1    11  6039 17002  4043  1521   462   503   218    50    86    22     4    15    19    19
   31   191   195   196   197   198   199   202   210   240   256   280   292   304   306   308   310
   22    14    61    95    13    35    12    14    18     9    23    24    21    26    17    19    28
  312   315   336   338   347   350   351   354   357   368   677
   30    17    22    19     9     9    33    26    24    20    21
```

```r
kobe_shot$day_difference[kobe_shot$day_difference > 7 | kobe_shot$day_difference < 1] <- 7
table(kobe_shot$day_difference)
```

```
    1     2     3     4     5     6     7
 6039 17002  4043  1521   462   503  1127
```

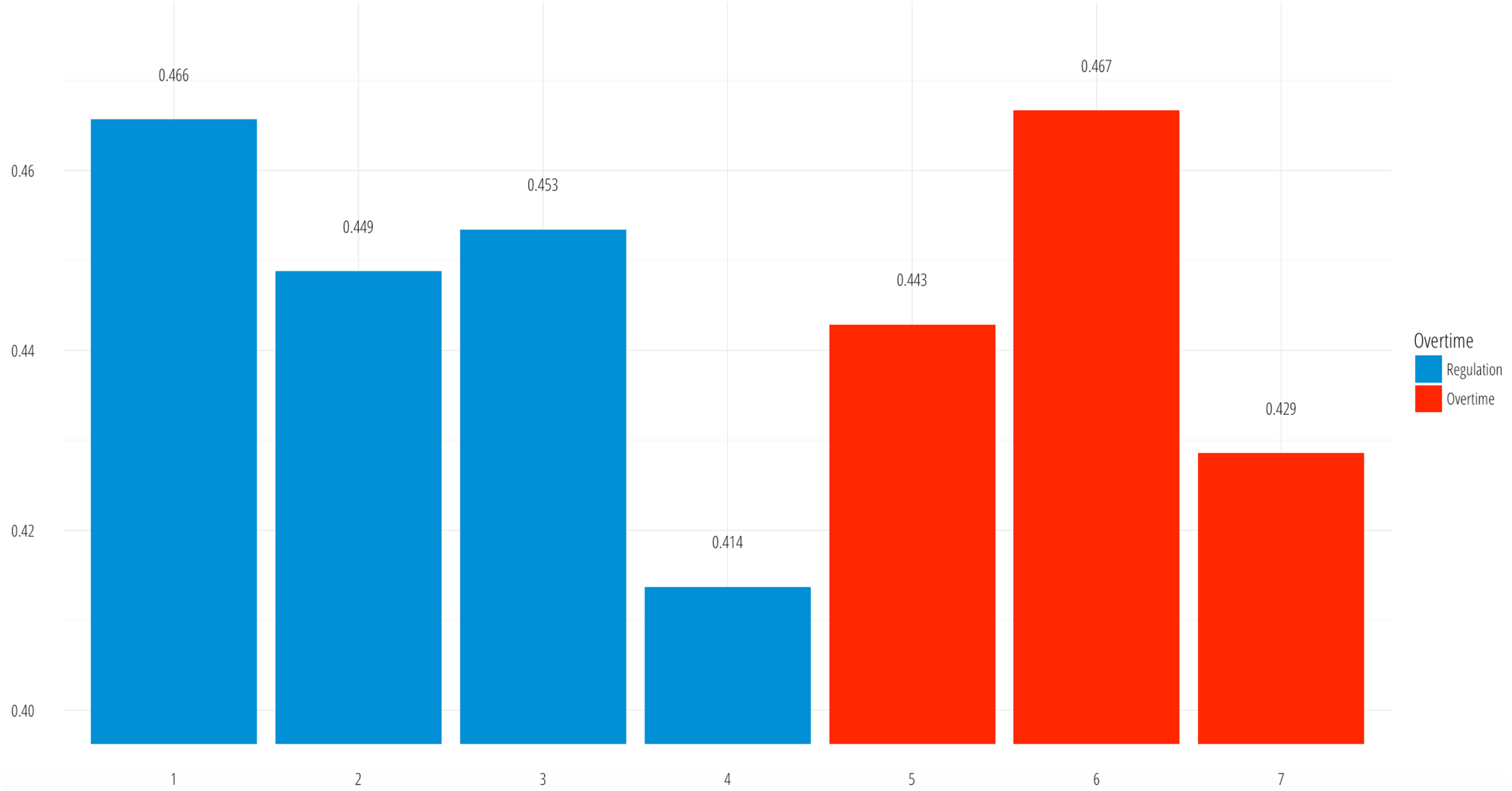
02 데이터 탐색/전처리



경기 쿼터별 시각화 (연장전 포함)

```
kobe_train %>%
        group_by(period) %>%
        summarise(rate = sum(shot_made_flag) / n(), count = n()) %>%
        mutate(overtime = period > 4) %>%
        ggplot(aes(x = factor(period), y = rate, fill = factor(overtime))) +
        geom_bar(stat = "identity") +
        scale_fill_fivethirtyeight("Overtime", labels = c("Regulation", "Overtime")) +
        coord_cartesian(ylim = c(0.4, 0.475)) +
        geom_text(aes(y = rate + 0.005, label = round(rate, 3)), size = 3.5,
                  family="OpenSans-CondensedLight")
```

02 데이터 탐색/전처리


0.466
0.449
0.453
0.414
0.443
0.467
0.429
0.46
0.44
0.42
0.40
1
2
3
4
5
6
7
Overtime
Regulation
Overtime

홈 & 어웨이 구분

```
> unique(kobe_shot$matchup)
 [1] LAL @ POR   LAL vs. UTA LAL @ VAN   LAL vs. LAC LAL @ HOU   LAL @ SAS   LAL vs. HOU LAL vs. DEN
 [9] LAL @ SAC   LAL @ DEN   LAL vs. CHI LAL vs. GSW LAL vs. MIN LAL @ LAC   LAL vs. IND LAL @ SEA
[17] LAL vs. SAS LAL vs. DAL LAL vs. PHI LAL @ GSW   LAL vs. SEA LAL vs. DET LAL vs. MIL LAL vs. VAN
[25] LAL @ TOR   LAL @ MIA   LAL @ DAL   LAL vs. POR LAL @ PHX   LAL vs. CLE LAL @ UTA   LAL vs. MIA
[33] LAL vs. NJN LAL @ NYK   LAL @ CLE   LAL @ MIN   LAL vs. CHH LAL vs. SAC LAL vs. PHX LAL @ NJN
[41] LAL @ PHI   LAL @ CHH   LAL @ IND   LAL vs. TOR LAL @ DET   LAL @ WAS   LAL @ ORL   LAL @ ATL
[49] LAL @ MIL   LAL vs. NYK LAL vs. MEM LAL vs. ORL LAL @ MEM   LAL @ CHI   LAL vs. WAS LAL vs. ATL
[57] LAL vs. BOS LAL @ BOS   LAL vs. NOH LAL @ NOH   LAL @ UTH   LAL vs. SAN LAL @ NOK   LAL @ PHO
[65] LAL vs. NOK LAL vs. PHO LAL @ CHA   LAL vs. CHA LAL vs. OKC LAL @ OKC   LAL vs. BKN LAL @ BKN
[73] LAL @ NOP   LAL vs. NOP
74 Levels: LAL @ ATL LAL @ BKN LAL @ BOS LAL @ CHA LAL @ CHH LAL @ CHI LAL @ CLE ... LAL vs. WAS
```

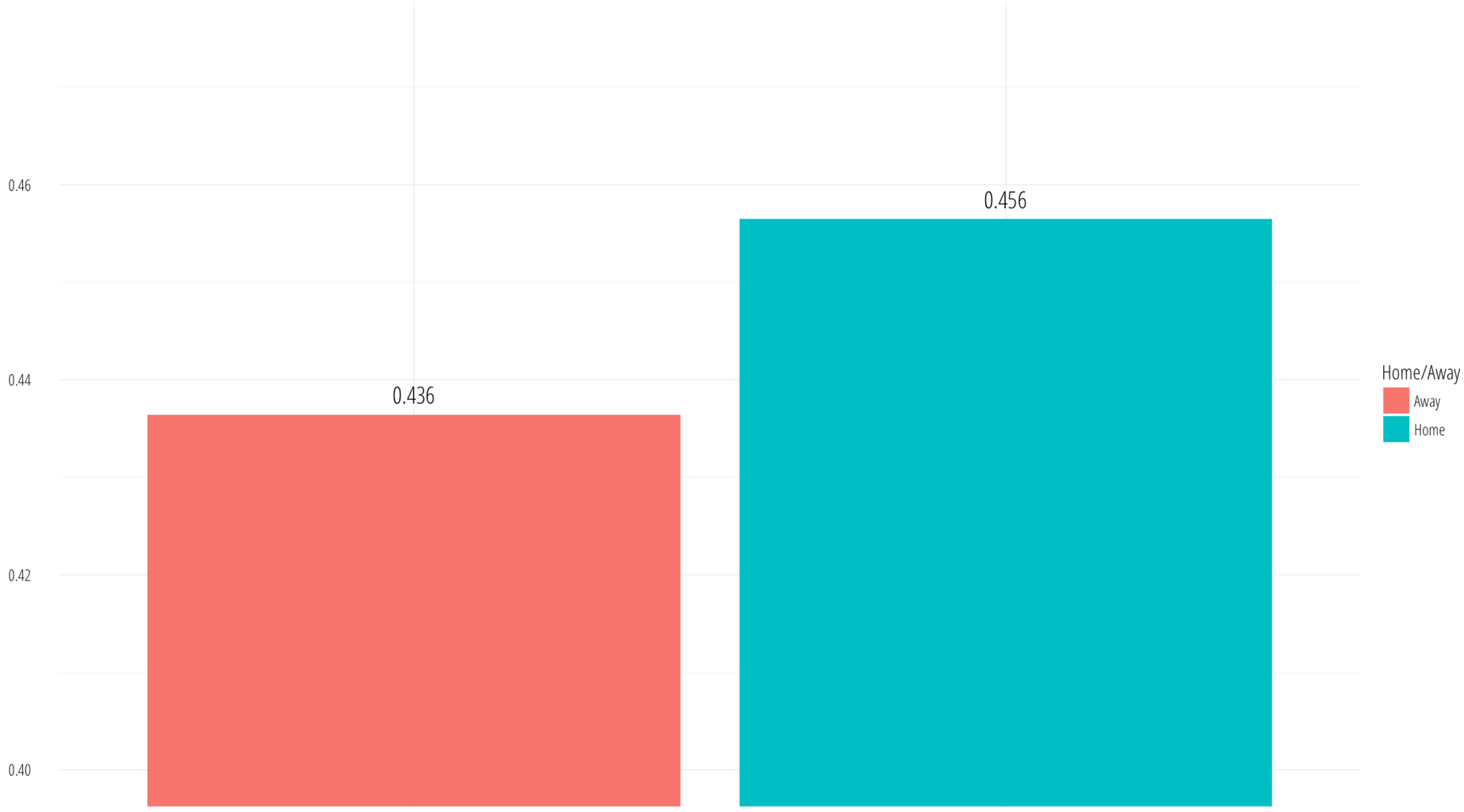
@ 이 포함되어 있으면 **어웨이** 경기

vs. 가 포함되어 있으면 **홈** 경기

홈 & 어웨이 구분

```
kobe_shot$home <- 1
kobe_shot$home[grep("@", kobe_shot$matchup)] <- 0

kobe_train %>%
        group_by(home) %>%
        summarise(rate = sum(shot_made_flag) / n()) %>%
        ggplot(aes(x = factor(home), y = rate, fill = factor(home))) +
        geom_bar(stat = "identity") +
        scale_fill_hue("Home/Away", labels = c("Away", "Home")) +
        coord_cartesian(ylim = c(0.4, 0.475)) +
        geom_text(aes(y = rate + 0.002, label = round(rate, 3)), size = 5,
                  family="OpenSans-CondensedLight") +
        theme(axis.text.x = element_blank())
```

02 데이터 탐색/전처리



[서고동저(西高東低)]

NBA에서 **서부**에 위치한 팀들이
동부의 팀들의 전력보다 더 강한 현상

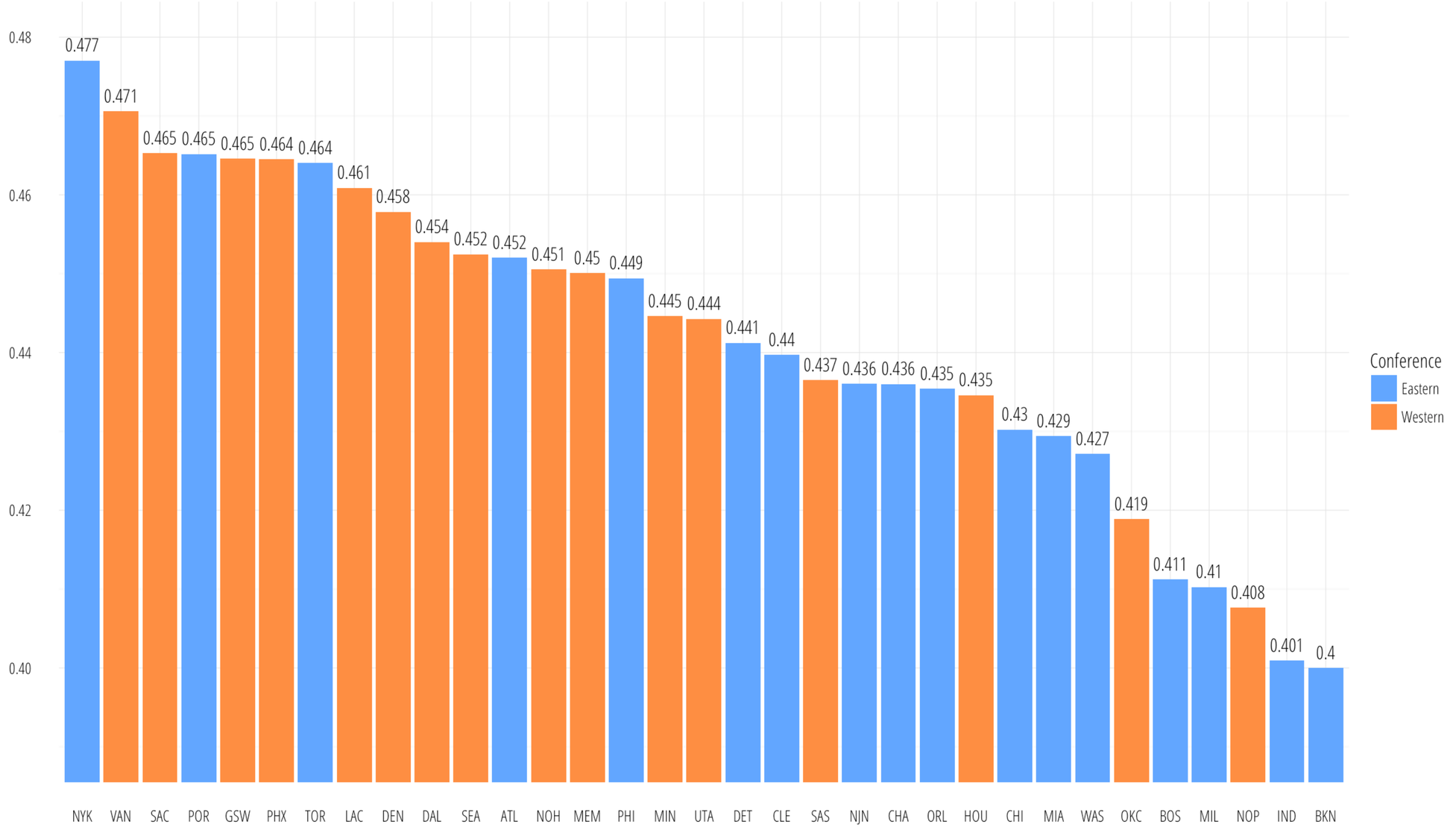

서고동저 해결?

마크 큐반의 컨퍼런스 개편안

서고동저. 한국지리 수업시간에 들을 수 있는 단어가 아니다. NBA에서 동부 컨퍼런스에 비해 서부 컨퍼런스가 지나치게 강한 현상을 나타내는 말이다. 시발점은 지난 2003-04시즌이었다. 당시 뉴욕 닉스 (39승 43패)와 보스턴 셀틱스 (36승 46패)는 저조한 성적에도 불구하고 동부 컨퍼런스 플레이오프에 진출했다. 반면, 서부 컨퍼런스에서는 5할 승률을 거둔 포틀랜드 트레일 블레이저스가 10위에 그치며 플레이오프에 탈락했다. 10년이 지난 지금, 서고동저 현상은 더욱 심화되었다. 글·남재우 사진·NBAE/Getty Images/멀티비츠

```
levels(kobe_shot$opponent)
conference <- c("Eastern", "Eastern", "Eastern", "Eastern", "Eastern", "Eastern",
                "Western", "Western", "Eastern", "Western", "Western", "Eastern",
                "Western", "Western", "Eastern", "Eastern", "Western", "Eastern",
                "Western", "Western", "Eastern", "Western", "Eastern", "Eastern",
                "Western", "Eastern", "Western", "Western", "Western", "Eastern",
                "Western", "Western", "Eastern")
kobe_shot$conference <- kobe_shot$opponent
levels(kobe_shot$conference) <- conference

kobe_train %>%
        group_by(opponent) %>%
        summarise(rate = sum(shot_made_flag) / n()) %>%
        mutate(conference = as.factor(conference)) %>%
        arrange(desc(rate)) %>%
        ggplot(aes(x = reorder(opponent, -rate), y = rate, fill = conference)) +
        geom_bar(stat = "identity") +
        coord_cartesian(ylim = c(0.39, 0.48)) +
        geom_text(aes(y = rate + 0.002, label = round(rate, 3)), size = 4,
                  family="OpenSans-CondensedLight") +
        scale_fill_manual("Conference", values = c("#62a6ff", "#ff8e41"))
```

02 데이터 탐색/전처리

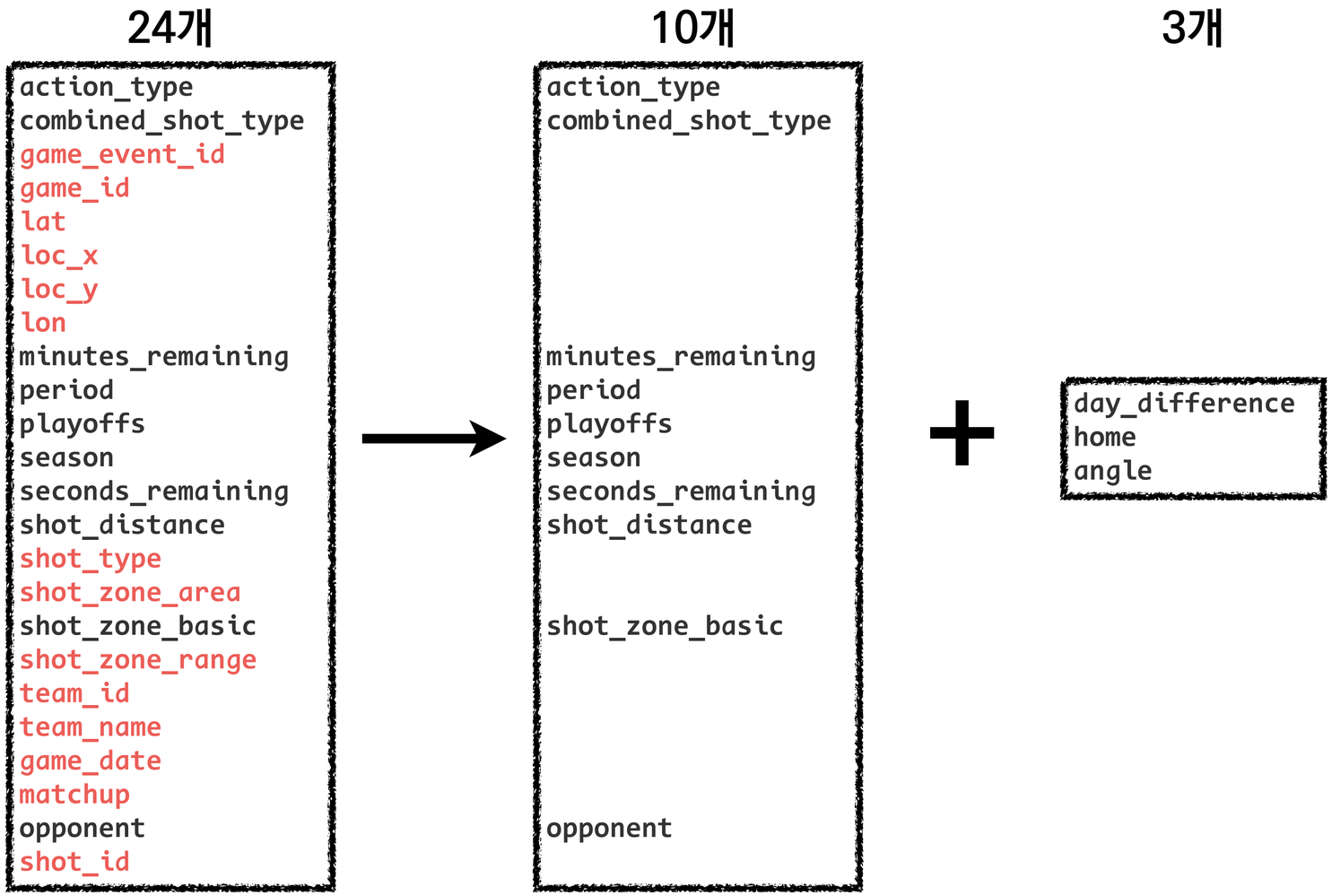

0.48
0.46
0.44
0.42
0.40
0.477
0.471
0.465 0.465 0.465 0.464 0.464
0.461
0.458
0.454
0.452 0.452
0.451
0.45
0.449
0.445 0.444
0.441
0.44
0.437 0.436 0.436 0.435 0.435
0.43
0.429
0.427
0.419
0.411 0.41
0.408
0.401
0.4
NYK VAN SAC POR GSW PHX TOR LAC DEN DAL SEA ATL NOH MEM PHI MIN UTA DET CLE SAS NJN CHA ORL HOU CHI MIA WAS OKC BOS MIL NOP IND BKN
Conference
Eastern
Western

슛을 쏜 위치는 **거리**와 **각도**에 따라 정해진다.

loc_x, loc_y, shot_zone_area, shot_zone_range 변수는
슛을 쏜 거리와 각도를 다시 학습하게 한다.

↓

거리 : shot_distance
각도 : angle?

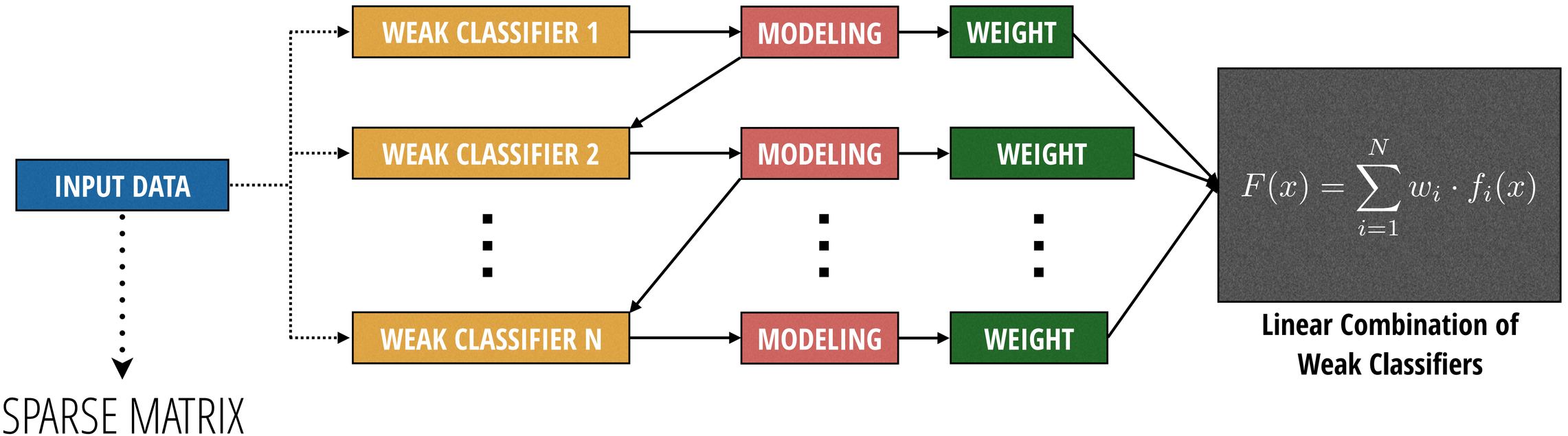

24개
action_type
combined_shot_type
game_event_id
game_id
lat
loc_x
loc_y
lon
minutes_remaining
period
playoffs
season
seconds_remaining
shot_distance
shot_type
shot_zone_area
shot_zone_basic
shot_zone_range
team_id
team_name
game_date
matchup
opponent
shot_id
10개
action_type
combined_shot_type
minutes_remaining
period
playoffs
season
seconds_remaining
shot_distance
shot_zone_basic
opponent
3개
day_difference
home
angle

03 예측 모델 구축

GRADIENT BOOSTING

기울기 강하(Gradient descent) + 부스팅 (Boosting)

GRADIENT BOOSTING



훈련 데이터의 분포를 바꿔가며
오분류된 결과에 초점을 맞춰 반복적으로 학습

GRADIENT BOOSTING

Machine Learning Challenge Winning Solutions

XGBoost is extensively used by machine learning practitioners to create state of art data science solutions, this is a list of machine learning winning solutions with XGBoost. Please send pull requests if you find ones that are missing here.

- Marios Michailidis, Mathias Müller and HJ van Veen, 1st place of the Dato Truely Native? competition. Link to the Kaggle interview.
- Vlad Mironov, Alexander Guschin, 1st place of the CERN LHCb experiment Flavour of Physics competition. Link to the Kaggle interview.
- Josef Slavicek, 3rd place of the CERN LHCb experiment Flavour of Physics competition. Link to the Kaggle interview.
- Mario Filho, Josef Feigl, Lucas, Gilberto, 1st place of the Caterpillar Tube Pricing competition. Link to the Kaggle interview.
- Qingchen Wang, 1st place of the Liberty Mutual Property Inspection. Link to the Kaggle interview.
- Chenglong Chen, 1st place of the Crowdflower Search Results Relevance. Link to the winning solution.
- Alexandre Barachant (“Cat”) and Rafał Cycoń (“Dog”), 1st place of the Grasp-and-Lift EEG Detection. Link to the Kaggle interview.
- Halla Yang, 2nd place of the Recruit Coupon Purchase Prediction Challenge. Link to the Kaggle interview.
- Owen Zhang, 1st place of the Avito Context Ad Clicks competition. Link to the Kaggle interview.
- Keiichi Kuroyanagi, 2nd place of the Airbnb New User Bookings. Link to the Kaggle interview.

- Marios Michailidis, Mathias Müller and Ning Situ, 1st place Homesite Quote Conversion. Link to the Kaggle interview.

LOGARITHMIC LOSS FUNCTION

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1 - y_i) \log(1 - p_i)].$$

N *: the number of actual value or predictive value*

y_i *: i_{th} actual value*

p_i *: i_{th} predictive value*

OVERFITTING
IN GRADIENT BOOSTING

그라디언트 부스팅은 **오분류된 결과**에 초점을 맞춰 학습한다.

↓

파라미터 튜닝 (Parameter Tuning)

DIVERGENCE
IN LOGARITHMIC LOSS FUNCTION

로그 손실 함수는 결과값이 **정반대**일 때, 값이 **발산**한다.

HOW TO AVOID OVERFITTING

Cross Validation

Reduce the Depth of Trees

Lower Learning Rate + More Iterations

Choose Important Features

STEP 1 MAKE SPARSE MATRIX

```
kobe_train_xgboost <- kobe_train_angle %>%
        select(action_type, combined_shot_type, period, minutes_remaining,
               seconds_remaining, playoffs, season, shot_distance,
               shot_zone_basic, opponent, day_difference, home, angle) %>%
        model.matrix(~., data = .)

kobe_test_xgboost <- kobe_test_angle %>%
        select(action_type, combined_shot_type, period, minutes_remaining,
               seconds_remaining, playoffs, season, shot_distance,
               shot_zone_basic, opponent, day_difference, home, angle) %>%
        model.matrix(~., data = .)
```

model.matrix()

데이터 프레임(Data Frame)을 희소행렬(Sparse Matrix)로 변환

STEP 2 Construct a Model

```
kobe_xgboost <- xgboost(data = kobe_train_xgboost, label = kobe_train_label,
                        max.depth = 7, eta = 0.04, subsample = 1,
                        nrounds = 105, objective = "binary:logistic",
                        eval_metric = "logloss")
```

PARAMETER	DESCRIPTION
data	모델 구축에 사용할 데이터 (희소행렬)
label	기존 데이터의 분류값 (shot_made_flag)
max.depth	트리의 최대 깊이 (깊을 수록 모델이 복잡해진다.)
eta	학습률(Learning rate)을 조절하는 변수 (값이 낮을 수록 과적합 해결)
subsample	학습 데이터의 서브샘플링 비율 (과적합 방지)
nrounds	모델의 반복 횟수 (값이 클 경우 과적합 위험)
objective	모델의 학습 목적
eval_metric	Validation Data의 평가 척도 결정

STEP 3 Parameter Tuning + Feature Selection

과적합 방지를 위한 **파라미터 튜닝**은 **필수**

1. 예측 모델에서 중요한 변수들을 확인
2. 모델을 최대한 단순하게 할 것
3. 학습률을 낮추고 반복 횟수를 높일 것

STEP 3 Parameter Tuning + Feature Selection

최초 답안 제출 시

총 17개 (기존 변수 13개 + 파생변수 4개)

학습률 0.3, 반복횟수 35회

로그 손실 함수값 : 0.61488 ⟶ **과적합 의심**

불필요한 변수들

`loc_x, loc_y, shot_type, shot_zone_area, overtime, clutch`

추가한 파생변수

`angle`

STEP 4 Make Predictions

```
1  submission <- predict(kobe_xgboost, kobe_test_xgboost)
```

```
> submission[1:100]
  [1] 0.3708518 0.4199390 0.7238642 0.7330666 0.3297993 0.4183267 0.4491391 0.4434871 0.7052602 0.3515517
 [11] 0.4066092 0.3328108 0.3400350 0.2070408 0.2822443 0.2997501 0.3932933 0.2973291 0.3104734 0.5665139
 [21] 0.5694830 0.3689822 0.3353061 0.3317943 0.7791578 0.6736338 0.2629861 0.3964520 0.7330666 0.4008136
 [31] 0.7301956 0.3884231 0.7873544 0.6925223 0.4031673 0.7199242 0.3538624 0.7120404 0.3375832 0.3788739
 [41] 0.2973369 0.4083768 0.9350109 0.3776821 0.3780594 0.3642684 0.6731837 0.7199242 0.3592673 0.3362932
 [51] 0.7977268 0.6796601 0.7593979 0.3640333 0.3488248 0.3167079 0.3465862 0.7575797 0.2829033 0.3735317
 [61] 0.7301956 0.7301956 0.7553506 0.3646601 0.3783795 0.2324837 0.5999032 0.4564244 0.3871702 0.3833336
 [71] 0.2849570 0.3936861 0.2375544 0.3457902 0.3642684 0.3860199 0.4013172 0.2783812 0.4223861 0.3841830
 [81] 0.3955283 0.7155708 0.3731934 0.3308206 0.3282012 0.3742516 0.3908551 0.3685691 0.5999032 0.4085901
 [91] 0.3288628 0.4518989 0.3468890 0.3153879 0.3619664 0.7216851 0.7667783 0.4240826 0.4035049 0.3836865
```

Knowledge • 233 teams

Kobe Bryant Shot Selection

Fri 15 Apr 2016 — Mon 13 Jun 2016 (39 days to go)

Dashboard

Public Leaderboard - Kobe Bryant Shot Selection

#	Δ5d	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	new	LeBronLearnsML *	0.59509	5	Mon, 02 May 2016 01:46:12
2	↑13	Dagar Katyal	0.59908	13	Wed, 04 May 2016 00:31:40 (-0.5h)
3	↓2	Jason Noriega	0.59915	28	Thu, 21 Apr 2016 04:13:59 (-26.9h)
4	↓2	Josh Stone	0.59934	1	Tue, 19 Apr 2016 18:24:57
5	↓2	Juan Aguilera	0.59953	5	Thu, 28 Apr 2016 02:11:47 (-9.2h)
6	↓2	anokas	0.60014	10	Thu, 28 Apr 2016 10:14:16 (-6.6d)
7	↓2	Jeff Mills	0.60024	48	Thu, 28 Apr 2016 09:25:35 (-10h)
8	**↓1**	**Jae-YoonHan**	**0.60064**	**56**	**Thu, 05 May 2016 13:50:21**
		⋮			
231	↓63	hntrwd	17.52519	1	Tue, 26 Apr 2016 05:05:22
232	↓63	ccdasme	17.66334	2	Tue, 26 Apr 2016 02:42:28 (-0.9h)
233	↓63	CharlesIrick	19.01747	1	Wed, 27 Apr 2016 00:41:06

04 고찰

RANDOM FOREST ?

공개된 스크립트의 대부분은 Python + Random Forest 조합

과적합 이슈를 해결할 수 있을까?

HOW TO IMPROVE