중간 보고서

NBA 선수에 대하여 분석하기

과 목	데이터 애널리틱스-R 기반 통계		
담당교수	강민형		
제출일	2020.10.26		
이름 (학과/학번/이름)	e 비즈니스학과	201823868	심소은
	e 비즈니스학과	201823869	조성우

중간 보고서

2020-10-26 심소은 조성우

목차

I. 서론

1.1 주제 선정 배경

- 포브스가 발표한 전 세계 2020 년 운동선수 수입이 가장 높은 순위에 따르면, 상위 100 명중 NBA(미국프로농구)가 35 명으로 가장 높습니다. 그 뒤를 이어 NFL(미국프로풋볼)이 31 명, 축구선수 14 명, 테니스 선수 6 명 순으로 나타났습니다. 고연봉을 자랑하는 선수들 중, 가장 많은 비율을 차지하는 농구선수. 그렇다면, 국내 농구선수들의 꿈의 무대라고 불리는 NBA 의 선수들은 어떤 특성을 지니고 있을지에 대한 궁금증이 들었기 때문에 선수들에 대하여 분석해보았습니다.

1.2 가설 설정

- 첫째로 NBA 에서 선수가 주로 사용하는 손에 따라 평균 연봉, 자유투 성공율, 드래프트 라운드에 차이를 보일 것이라는 가설을 설정했습니다.
- 두번째로 선수의 Stat 에 따라 연봉에 차이를 보일 것이라는 가설을 설정했습니다. 상위 연봉 25%그룹과 하위 연봉 25% 그룹의 승리기여도 stat 과 포인트 득점 stat 을 변수로 설정했습니다.

1.3 사용패키지

분석에 활용한 패키지 목록입니다.

```
library(dplyr)

library(measurements)
library(descr)
library(ggplot2)
library(RColorBrewer)
library(sjPlot)

library(ggpubr)
```

II. 본론

1. 데이터 소개 및 전처리

1.1 데이터 소개

- 'players.csv' 'players.csv' 에는 선수들의 인적사항 정보와 career 동안 성적, 출신대학, 출신고교, 키, 몸무게, 포지션, 드래프트 정보 등이 포함되어 있습니다.

- 'salaries_1985to2018.csv' 'salaries_1985to2018.csv' 에는 'players.csv'에 할당된 player_id 에 따른 선수별 시즌 연봉 데이터가 포함되어 있습니다.

저희는 프로젝트의 분석을 NBA 공식홈페이지의 데이터를 2 차 가공한 secondary data 를 분석에 활용했으며, 데이터의 출처는 http://data.world 입니다.

1.2 데이터 전처리

- df 'players.csv'와 'salaries_1985to2018.csv'을 'player_id' 변수 기준으로 inner_join 한 데이터프레임 df 생성

```
player_df = read.csv('players.csv',stringsAsFactors = F)
salaries_df = read.csv('salaries_1985to2018.csv')

colnames(player_df[1]) = 'player_id'

df = inner_join(player_df,salaries_df,by=c('X_id'='player_id'))
```

- df2 선수들의 시즌별 연봉이 아닌 선수별 연봉평균을 분석에 활용하기 위한 데이터프레임

```
average_salary =aggregate(salary~X_id,df,mean)
df2 = inner_join(player_df,average_salary,by='X_id')
```

- 체중 변수 전처리 데이터 중 체중의 경우 lb 의 단위로 적용되어 있어 character 값으로 저장되어 summary 를 활용하기 제한이 되고, "lbs"보다 "kg"가 더욱 직관적으로 활용하기 용이하므로 단위를 kg 로 변환해 저장합니다.

```
df2$weight = gsub("lb",'',df2$weight)
df2$weight <- as.numeric(df2$weight)
df2$weight <- conv_unit(df2$weight, "lbs", "kg")
```

- stat 변수 전처리 데이터 중 numeric 변환이 필요한 변수의 변환

```
df2$career_WS <- as.numeric(df2$career_WS)
df2$career_FG. <- as.numeric(df2$career_FG.)

df2$career_FG3. <- as.numeric(df2$career_FG3.)

df2$career_FT. <- as.numeric(df2$career_FT.)

df2$career_PER <- as.numeric(df2$career_PER)

df2$career_TRB <- as.numeric(df2$career_TRB)
df2$career_eFG. <- as.numeric(df2$career_eFG.)
```

2. 기술통계 및 시각화

2.1 변수 별 요약기술통계 및 시각화

- 2.1.1 연봉 통계

```
summary(df2$salary)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     2706   349250   931784  1964587  2535453 16411903
```

가장 연봉이 높은 선수

```
df2$name[which.max(df2$salary)]

## [1] "Kobe Bryant"
```

가장 연봉이 낮은 선수

```
df2$name[which.min(df2$salary)]

## [1] "Jason Sasser"
```

- 2.1.2 신장 통계

```
summary(df2$height)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   160.0   193.0   201.0   200.8   208.0   231.0
```

가장 키가 큰 선수

```
df2$name[which.max(df2$height)]

## [1] "Manute Bol"
```

가장 키가 작은 선수

```
df2$name[which.min(df2$height)]
```

```
## [1] "Muggsy Bogues"
```

- 2.1.3 체중 통계

```
summary(df2$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.33   88.45   97.98   98.61  107.05  163.29
```

가장 체중이 높은 선수

```
df2$name[which.max(df2$weight)]
```

```
## [1] "Sim Bhullar"
```

가장 체중이 낮은 선수

```
df2$name[which.min(df2$weight)]
```

```
## [1] "Spud Webb"
```

- 2.1.4 출신 대학교 통계 가장 많은 NBA 선수를 배출한 대학교 Top5 는 어디인가?

```
df2$college <- as.factor(df2$college)
head(summary(df2$college))
```

```
##                                                University of Kentuck
y
##                                       269                                    5
9
## University of California, Los Angeles          University of North Carolin
a
##                                        58                                    5
2
##                           Duke University                  University of Kansa
s
##                                        49                                    4
4
```

- 2.1.5 출신 고등학교 통계 가장 많은 NBA 선수를 배출한 고등학교 Top5 는 어디인가?

```
df2$highSchool <- as.factor(df2$highSchool)
head(summary(df2$highSchool))
```

```
##
##                                                208
##  Oak Hill Academy in Mouth of Wilson, Virginia
##                                                 28
## Hargrave Military Academy in Chatham, Virginia
##                                                 13
##   Brewster Academy in Wolfeboro, New Hampshire
```

```
##                                                 9
##              Proviso East in Maywood, Illinois
##                                                 9
##             Mater Dei in Santa Ana, California
##                                                 8
```

- 2.1.6 출신지 통계 가장 많은 NBA 선수를 배출한 지역 Top 10 은 어디인가?

```
df2$birthPlace <- as.factor(df2$birthPlace)
head(summary(df2$birthPlace),10)
```

```
##                Chicago, Illinois          Los Angeles, California
##                               79                               69
##       Philadelphia, Pennsylvania                Detroit, Michigan
##                               46                               37
## Washington, District of Columbia               Brooklyn, New York
##                               37                               34
##              Baltimore, Maryland                 Atlanta, Georgia
##                               31                               29
##                   Houston, Texas                    Dallas, Texas
##                               27                               26
```

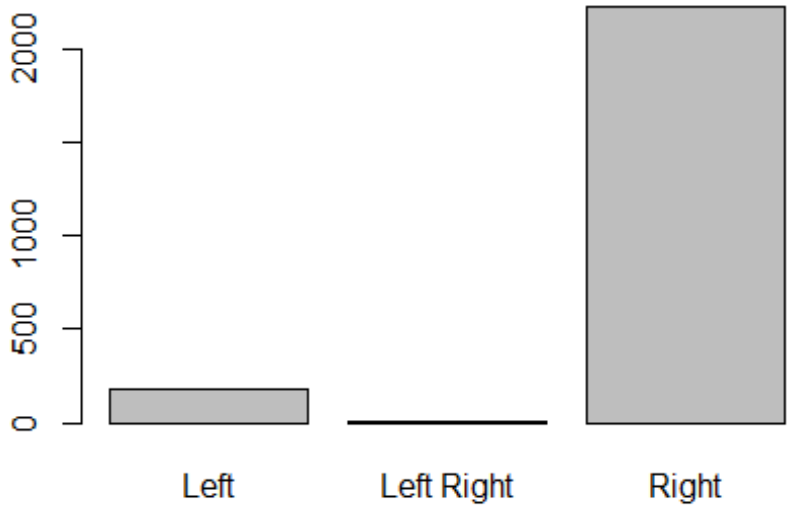
- 2.1.7 포지션 통계 NBA 에 소속된 선수들의 포지션 분포

```
head(sort(table(df2$position), decreasing = TRUE),13)
```

```
##
##                           Center                      Point Guard
##                              374                              332
##                   Shooting Guard                     Power Forward
##                              307                              271
##                    Small Forward          Power Forward and Center
##                              256                              124
##         Center and Power Forward    Point Guard and Shooting Guard
##                              120                              102
##   Shooting Guard and Point Guard Small Forward and Shooting Guard
##                              100                               97
## Shooting Guard and Small Forward  Power Forward and Small Forward
##                               92                               89
##  Small Forward and Power Forward
##                               76
```

- 2.1.8 슈팅에 사용하는 주 손 통계 왼손, 오른손 분포

```
freq(df2$shoots)
```

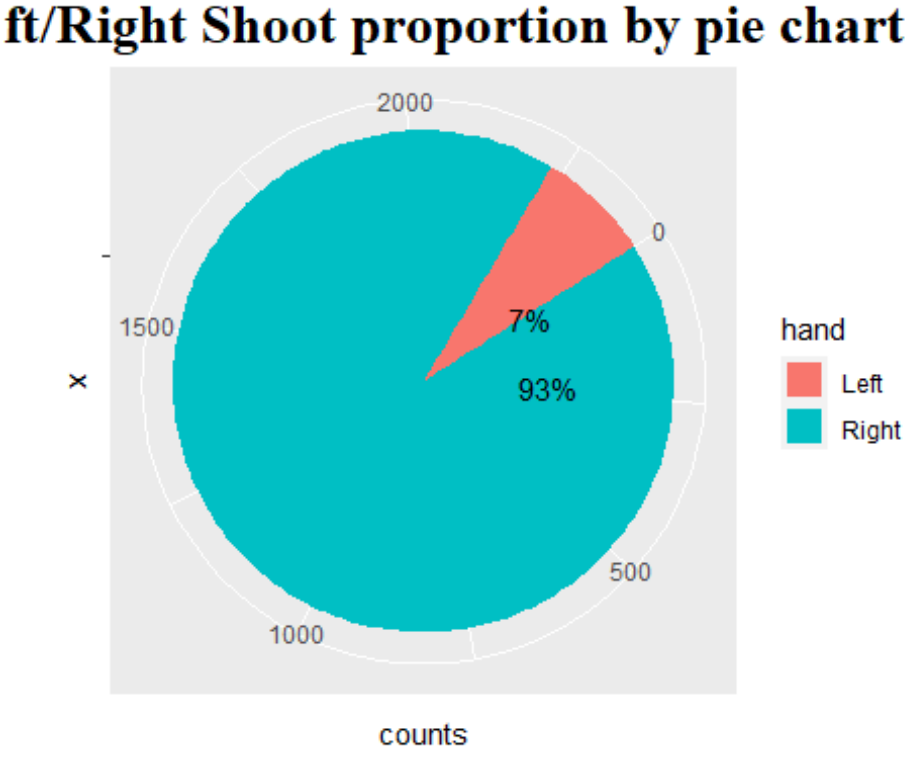



```
## df2$shoots
##            Frequency   Percent
## Left             177   7.35050
## Left Right         1   0.04153
## Right           2230  92.60797
## Total           2408 100.00000
```

```
df_table = data.frame(hand=c('Left','Right'),counts=c(177,2230))
df_table$prop = c((177/(177+2230))*100,(2230/(177+2230))*100)
df_table$prop = round(df_table$prop)

df_table
```

```
##    hand counts prop
## 1  Left    177    7
## 2 Right   2230   93
```

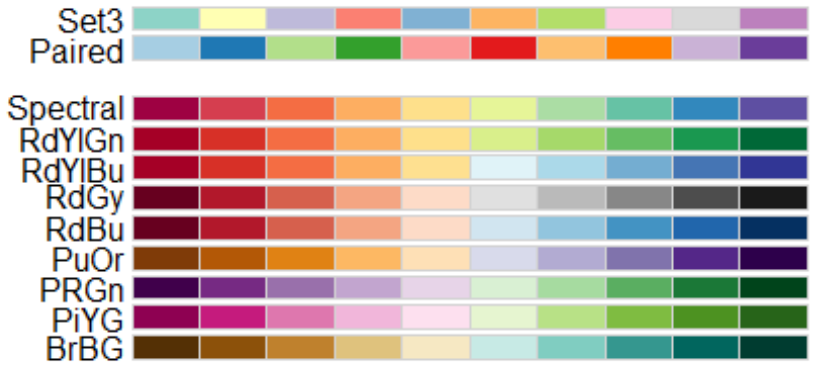
```
ggplot(data=df_table)+
  geom_bar(aes(x='',y=counts,fill=hand),width=4,size=1.5,stat='identity')+
  coord_polar('y',start=1)+
  ggtitle('Left/Right Shoot proportion by pie chart')+
  geom_text(aes(x='',y=prop/2,label=paste0(prop,"%")),position=position_stack
(vjust=5))+
  theme(plot.title = element_text(family='serif',face="bold",hjust=0.5,size=2
0,color='Black'))
```



- 2.1.9 시즌에 따른 NBA 평균연봉 변화

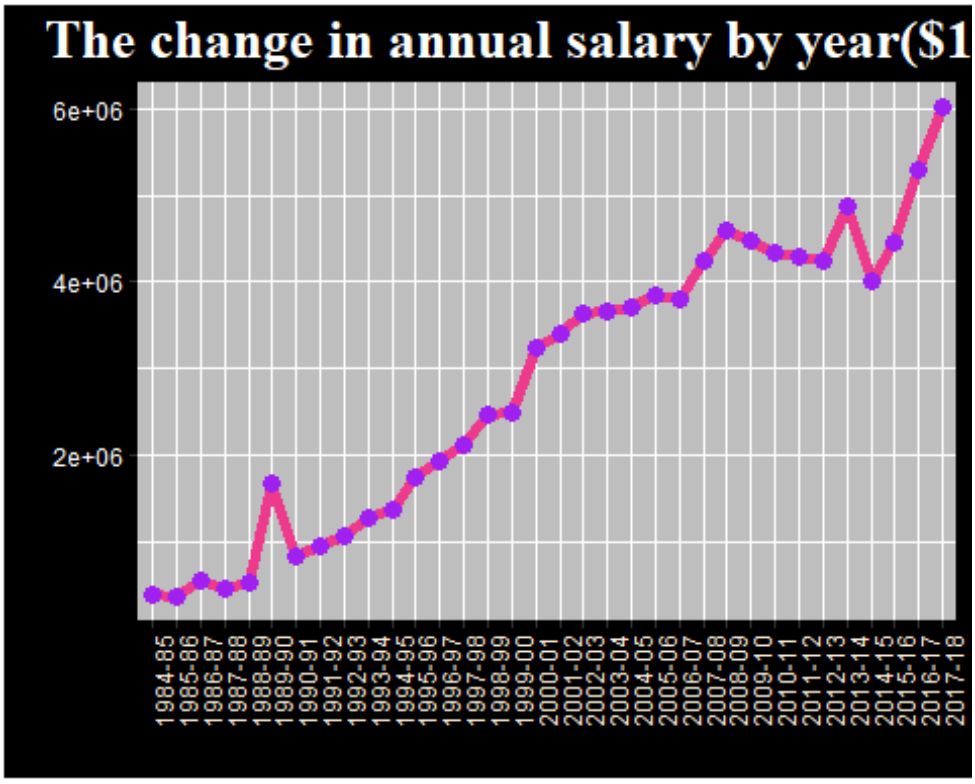
```
ss = data.frame(df$salary,df$season)
ss = aggregate(ss,list(ss$df.season),mean)

colnames(ss) = c('SEASON','AVG_salary')
ss$AVG_salary = round(ss$AVG_salary)
ss <- ss[-3]
ss

##     SEASON AVG_salary
## 1  1984-85     398805
## 2  1985-86     370103
## 3  1986-87     543033
## 4  1987-88     459204
## 5  1988-89     528011
## 6  1989-90    1670938
## 7  1990-91     831623
## 8  1991-92     954346
## 9  1992-93    1061758
## 10 1993-94    1269884
## 11 1994-95    1360404
## 12 1995-96    1734345
## 13 1996-97    1935172
## 14 1997-98    2121415
## 15 1998-99    2456729
```

```
## 16 1999-00    2498708
## 17 2000-01    3236011
## 18 2001-02    3395648
## 19 2002-03    3636158
## 20 2003-04    3650914
## 21 2004-05    3705740
## 22 2005-06    3839337
## 23 2006-07    3793016
## 24 2007-08    4244496
## 25 2008-09    4581769
## 26 2009-10    4476258
## 27 2010-11    4334884
## 28 2011-12    4289844
## 29 2012-13    4239567
## 30 2013-14    4865642
## 31 2014-15    4018814
## 32 2015-16    4441493
## 33 2016-17    5282659
## 34 2017-18    6011008
```




```
ggplot(data=ss,aes(x=SEASON,y=AVG_salary))+
  geom_line(data=ss,aes(x=SEASON,y=AVG_salary,group=1),color='violetred2',siz
e=2) +
  geom_point(data=ss,aes(x=SEASON,y=AVG_salary,group=1),color='purple',size=
3)+
  ggtitle('The change in annual salary by year($100)')+
```

```
  theme(plot.title = element_text(family='serif',face="bold",hjust=0.5,size=2
0,color='White'),plot.subtitle =element_text(vjust=1),plot.caption=element_te
xt(vjust=1),axis.text.x=element_text(angle=90,color='antiquewhite'),axis.tex
t.y=element_text(color='white')) + theme(panel.background = element_rect(fill
='grey'),plot.background = element_rect(fill='black'))
```




```
# 시즌의 흐름에 따라 선수들이 받는 연봉은 꾸준히 우상향하는 모습을 보여주고 있습니다.
```

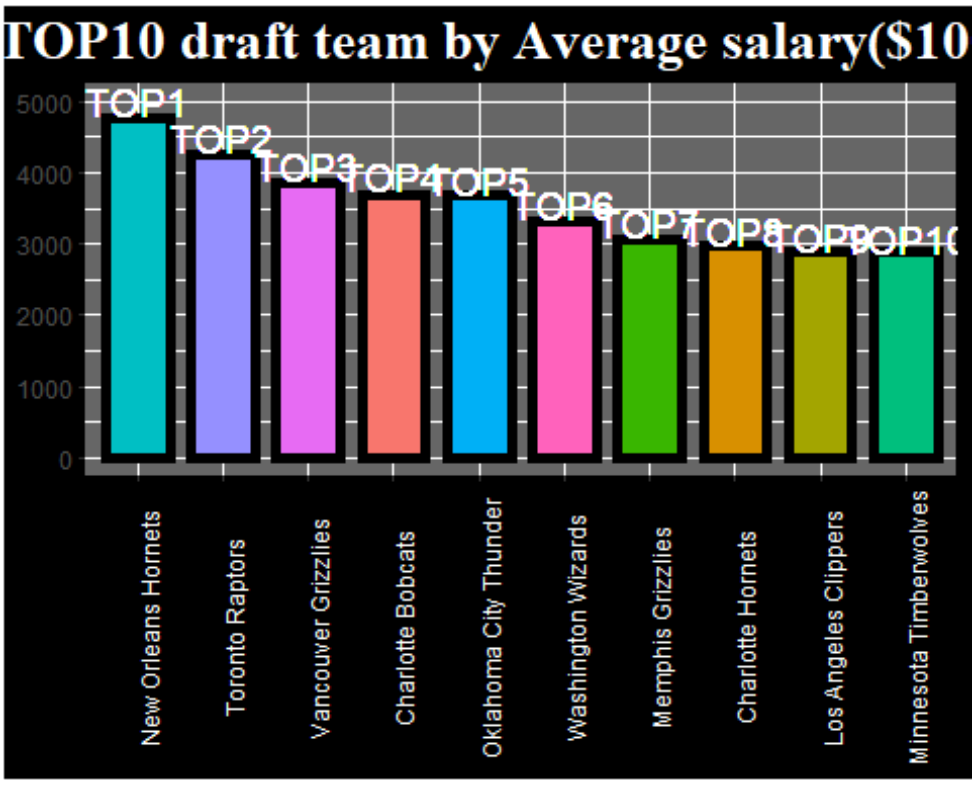
- 2.1.10 드래프트 팀별 연봉 TOP 10

```
draft_mean = aggregate(salary~draft_team,df2,mean)
draft_mean = draft_mean[order(-draft_mean$salary),]
rownames(draft_mean) = NULL

draft_mean_top = draft_mean[1:10,]
draft_mean_top
```

```
##                draft_team  salary
## 1     New Orleans Hornets 4783938
## 2         Toronto Raptors 4264622
## 3     Vancouver Grizzlies 3870633
## 4       Charlotte Bobcats 3687932
## 5   Oklahoma City Thunder 3683332
```

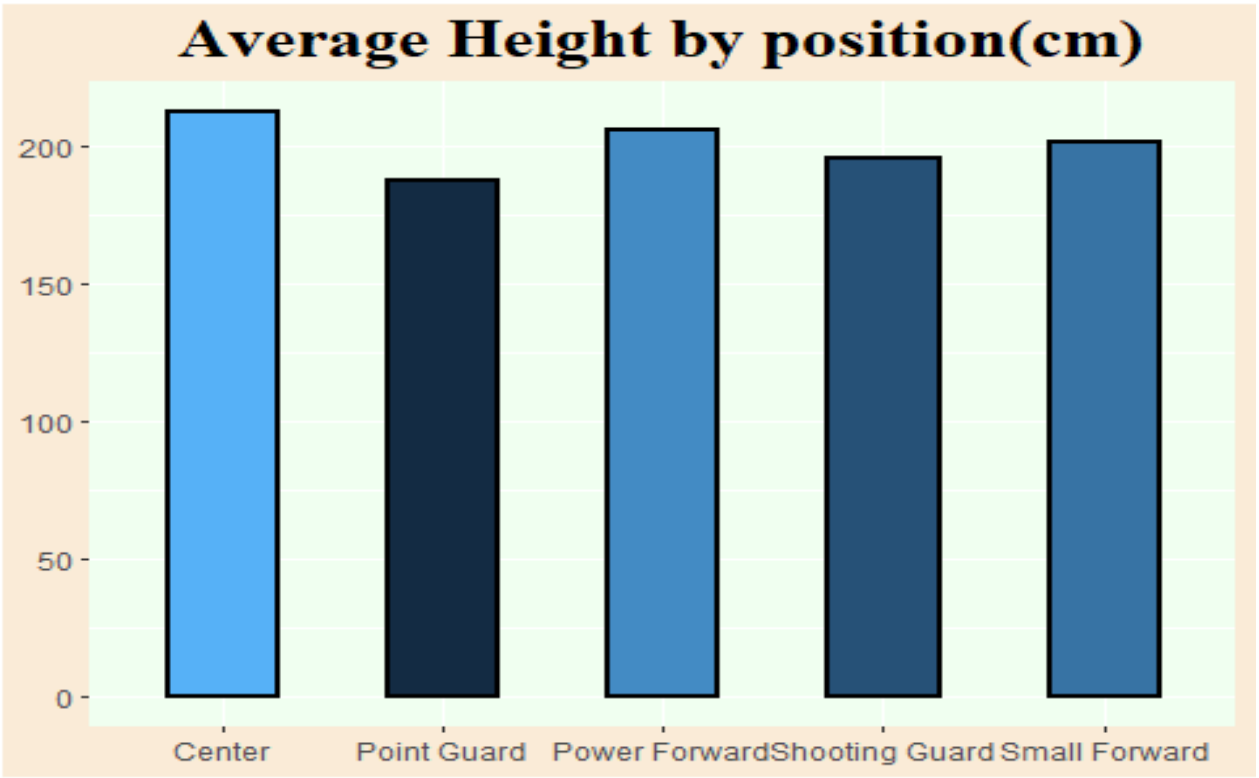
```
## 6      Washington Wizards 3313439
## 7       Memphis Grizzlies 3061711
## 8       Charlotte Hornets 2974693
## 9    Los Angeles Clippers 2892459
## 10 Minnesota Timberwolves 2884676
```

```
ggplot(draft_mean_top,aes(reorder(draft_team,-salary/1000),salary/1000,fill=draft_team))+
  geom_bar(width=0.75,stat = 'identity',colour='black',size=2)+
  xlab("")+ylab("")+ggtitle('TOP10 draft team by Average salary($1000)') + theme(plot.title= element_text(family='serif',face="bold",hjust=0.5,size=20,color='White'),legend.position='none') +labs(x=NULL,y=NULL) + theme(plot.subtitle =element_text(vjust=1),plot.caption=element_text(vjust=1),axis.text.x=element_text(angle=90,color='white'))+
  geom_text(aes(x=1,y=5000,label='TOP1'),color='white',size=5)+
  geom_text(aes(x=2,y=4500,label='TOP2'),color='white',size=5)+
  geom_text(aes(x=3,y=4100,label='TOP3'),color='white',size=5)+
  geom_text(aes(x=4,y=3950,label='TOP4'),color='white',size=5)+
  geom_text(aes(x=5,y=3900,label='TOP5'),color='white',size=5)+
  geom_text(aes(x=6,y=3550,label='TOP6'),color='white',size=5)+
  geom_text(aes(x=7,y=3300,label='TOP7'),color='white',size=5)+
  geom_text(aes(x=8,y=3200,label='TOP8'),color='white',size=5)+
  geom_text(aes(x=9,y=3100,label='TOP9'),color='white',size=5)+
  geom_text(aes(x=10.01,y=3080,label='TOP10'),color='white',size=5)+
  theme(panel.background = element_rect(fill='grey40'),plot.background = element_rect(fill='black'))
```



- 2.1.11 체급에 따른 포지션 분포 포지션 별 평균 신장

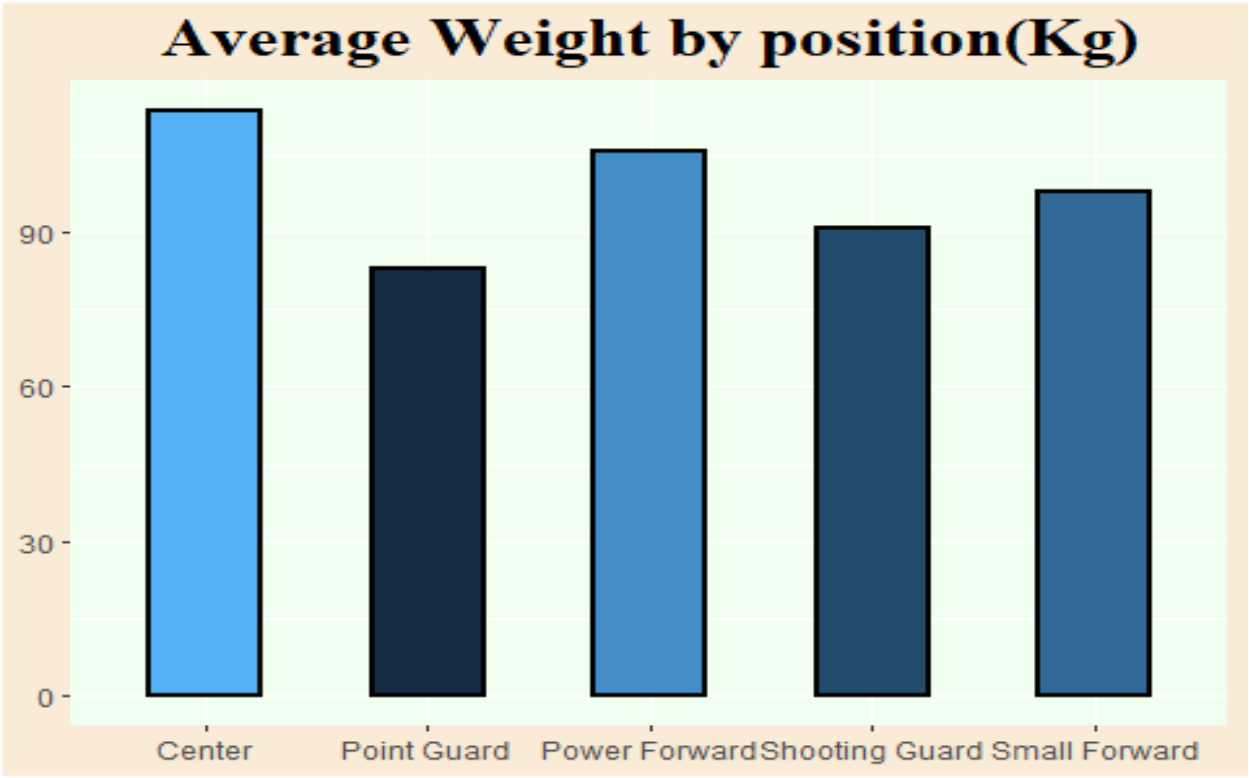
```
df_hp <- data.frame(df2$position,df2$height)


df_hp_avg <- aggregate(df_hp,list(df_hp$df2.position),mean)

df_hp_avg <- df_hp_avg[-2]
colnames(df_hp_avg)= c('position','height_avg')
df_hp_avg1 = df_hp_avg[c(1,5,11,19,27),]

ggplot(df_hp_avg1,aes(position,height_avg,fill=height_avg))+
  geom_bar(width=0.50,stat = 'identity',colour='black',size=1)+
  xlab("")+ylab("")+ggtitle('Average Height by position(cm)') +
  theme(legend.position='none') +
  labs(x=NULL,y=NULL) +
  theme(plot.title = element_text(family='serif',face="bold",hjust=0.5,size=2
0,color='Black'))+
  theme(panel.background = element_rect(fill='honeydew1'),plot.background = e
lement_rect(fill='antiquewhite'))
```



포지션 별 평균 체중

```
df_wp <- data.frame(df2$position,df2$weight)
```

```r
df_wp_avg <- aggregate(df_wp,list(df_wp$df2.position),mean)

df_wp_avg <- df_wp_avg[-2]
colnames(df_wp_avg)= c('position','weight_avg')
df_wp_avg1 = df_wp_avg[c(1,5,11,19,27),]


ggplot(df_wp_avg1,aes(position,weight_avg,fill=weight_avg))+
  geom_bar(width=0.50,stat = 'identity',colour='black',size=1)+
  xlab("")+ylab("")+ggtitle('Average Weight by position(Kg)') +
  theme(legend.position='none') +
  labs(x=NULL,y=NULL) +
  theme(plot.title = element_text(family='serif',face="bold",hjust=0.5,size=2
0,color='Black'))+
  theme(panel.background = element_rect(fill='honeydew1'),plot.background = e
lement_rect(fill='antiquewhite'))
```



- 2.1.12 선수별 stat 통계

career_AST : Assists

```r
summary(df2$career_AST)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.500   1.100   1.551   2.100  11.200

df2$name[which.max(df2$career_AST)]
```

```
## [1] "Magic Johnson"
```

```
df2$name[which.min(df2$career_AST)]
```

```
## [1] "Cliff Alexander"
```

career_FG. : Field Goal Percentage

```
summary(df2$career_FG.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   40.50   44.00   43.59   47.90  100.00       9
```

```
df2$name[which.max(df2$career_FG.)]
```

```
## [1] "Trey Gilder"
```

```
df2$name[which.min(df2$career_FG.)]
```

```
## [1] "Martynas Andriuskevicius"
```

career_FG3. : 3-Point field goal Percentage

```
summary(df2$career_FG3.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    16.7    29.8    25.4    35.0   100.0     219
```

```
df2$name[which.max(df2$career_FG3.)]
```

```
## [1] "Eric Anderson"
```

```
df2$name[which.min(df2$career_FG3.)]
```

```
## [1] "Alaa Abdelnaby"
```

career_FT. : Free Throw Percentage

```
summary(df2$career_FT.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   65.80   73.70   71.45   79.30  100.00      69
```

```
df2$name[which.max(df2$career_FT.)]
```

```
## [1] "Keith Appling"
```

```
df2$name[which.min(df2$career_FT.)]
```

```
## [1] "William Cunningham"
```

career_G : Games

```
summary(df2$career_G)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    69.0   255.0   367.6   612.0  1611.0
```

```
df2$name[which.max(df2$career_G)]
```

```
## [1] "Robert Parish"
```

```
df2$name[which.min(df2$career_G)]
```

```
## [1] "JamesOn Curry"
```

career_PTS : Points

```
summary(df2$career_PTS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.200   5.800   6.993   9.500  30.100
```

```
df2$name[which.max(df2$career_PTS)]
```

```
## [1] "Michael Jordan"
```

```
df2$name[which.min(df2$career_PTS)]
```

```
## [1] "Martynas Andriuskevicius"
```

career_PER : Player Efficiency Rating (A measure of per-minute production standardized such that the league average is 15.)

```
summary(df2$career_PER)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -48.60    9.80   12.20   12.03   14.60   88.30       2
```

```
df2$name[which.max(df2$career_PER)]
```

```
## [1] "Steven Hill"
```

```
df2$name[which.min(df2$career_PER)]
```

```
## [1] "Mile Ilic"
```

career_TRB : Total Rebounds

```
summary(df2$career_TRB)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.50    2.60    3.09    4.10   13.70
```

```
df2$name[which.max(df2$career_TRB)]
```

```
## [1] "Andre Drummond"
```

```
df2$name[which.min(df2$career_TRB)]
```

```
## [1] "JamesOn Curry"
```

career_WS : Win shares (An estimate of the number of wins contributed by a player.)
승리에 기여한 횟수

```
summary(df2$career_WS)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -2.70    0.30    5.60   19.12   26.00  273.40

df2$name[which.max(df2$career_WS)]

## [1] "Kareem Abdul-Jabbar"

df2$name[which.min(df2$career_WS)]

## [1] "Lancaster Gordon"
```

career_eFG : Effective Field Goal Percentage (This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal.)

```
summary(df2$career_eFG.)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   43.90   47.50   46.47   50.50  150.00       9

df2$name[which.max(df2$career_eFG.)]

## [1] "Jordan Sibert"

df2$name[which.min(df2$career_eFG.)]

## [1] "Martynas Andriuskevicius"
```

3. 가설검증 추론통계

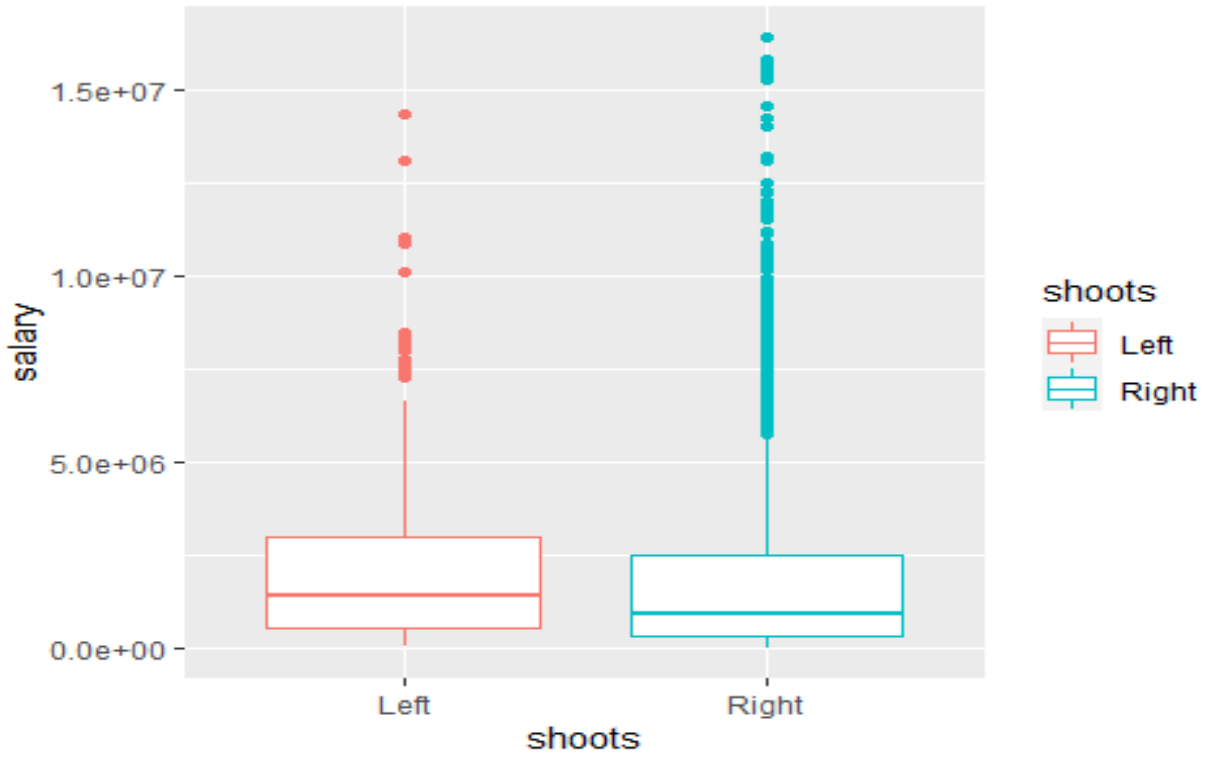
3.1 주 사용 손에 따른 추론통계

- 3.1.1 왼손 선수집단과 오른손 선수집단의 평균연봉에는 차이가 있을 것이다.

```
shoots_r = subset(df2,shoots=='Right')
shoots_l = subset(df2,shoots=='Left')
shoots_all =rbind(shoots_r,shoots_l)
```

- Two sample t-test

```
ggplot(shoots_all,aes(x=shoots,y=salary,col=shoots))+
  geom_boxplot()
```

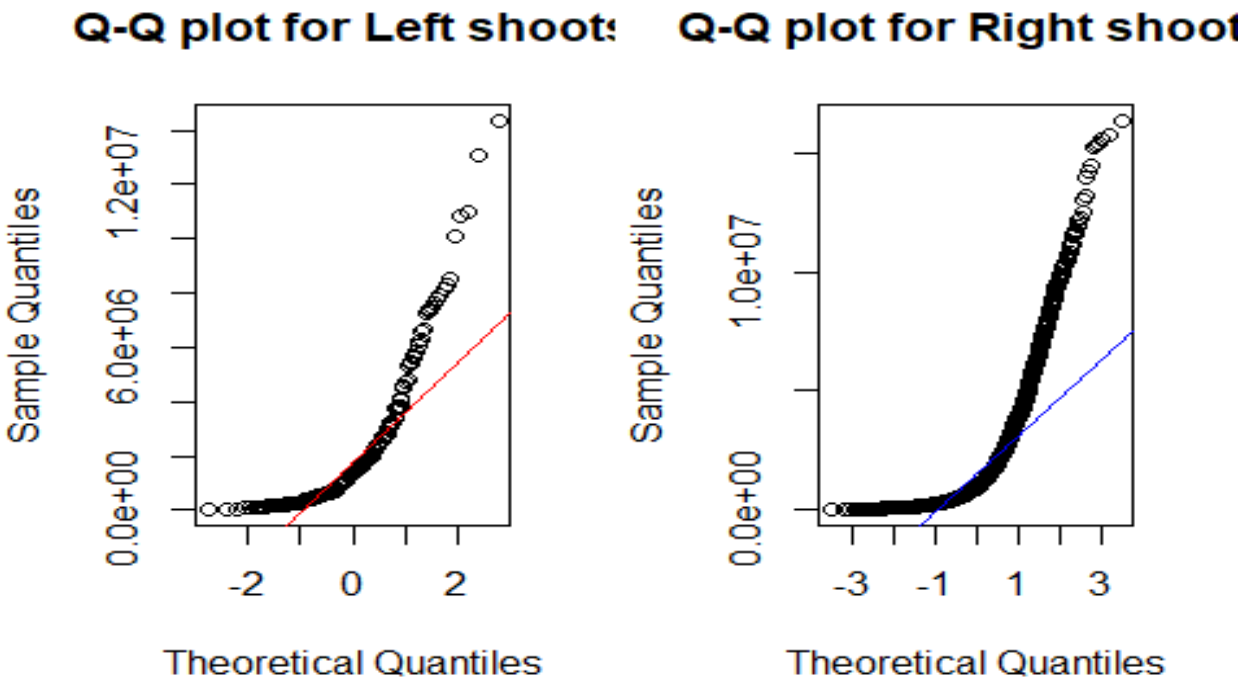


등분산 검정 var.test

```
var.test(salary~shoots,data=shoots_all)

##
##  F test to compare two variances
##
## data:  salary by shoots
## F = 1.149, num df = 176, denom df = 2229, p-value = 0.19
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9338156 1.4430136
## sample estimates:
## ratio of variances
##            1.14902
```

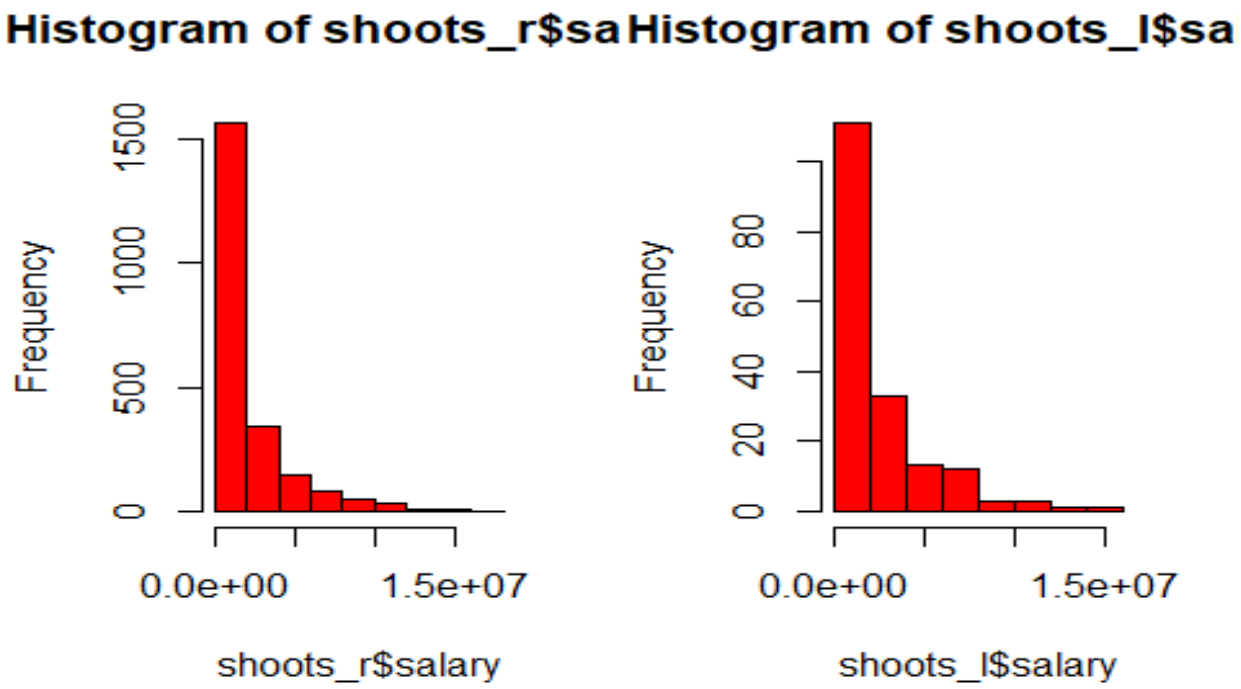
정규성 검정(1) qqplot

```
par(mfrow=c(1,2))
qqnorm(shoots_l$salary,main="Q-Q plot for Left shoots");qqline(shoots_l$salar
y,col='red')
qqnorm(shoots_r$salary,main="Q-Q plot for Right shoots");qqline(shoots_r$sala
ry,col='blue')
```



정규성 검정(2) plot histogram

```
par(mfrow=c(1,2))
hist(shoots_r$salary,breaks=10,col=2)
hist(shoots_l$salary,breaks=10,col=2)
```



정규성 검정(3) shapiro test

```
options(scipen=999)
shapiro.test(shoots_l$salary)

##
##  Shapiro-Wilk normality test
##
## data:  shoots_l$salary
## W = 0.77143, p-value = 0.000000000000002487

shapiro.test(shoots_r$salary)

##
##  Shapiro-Wilk normality test
##
## data:  shoots_r$salary
## W = 0.72055, p-value < 0.00000000000000022
```

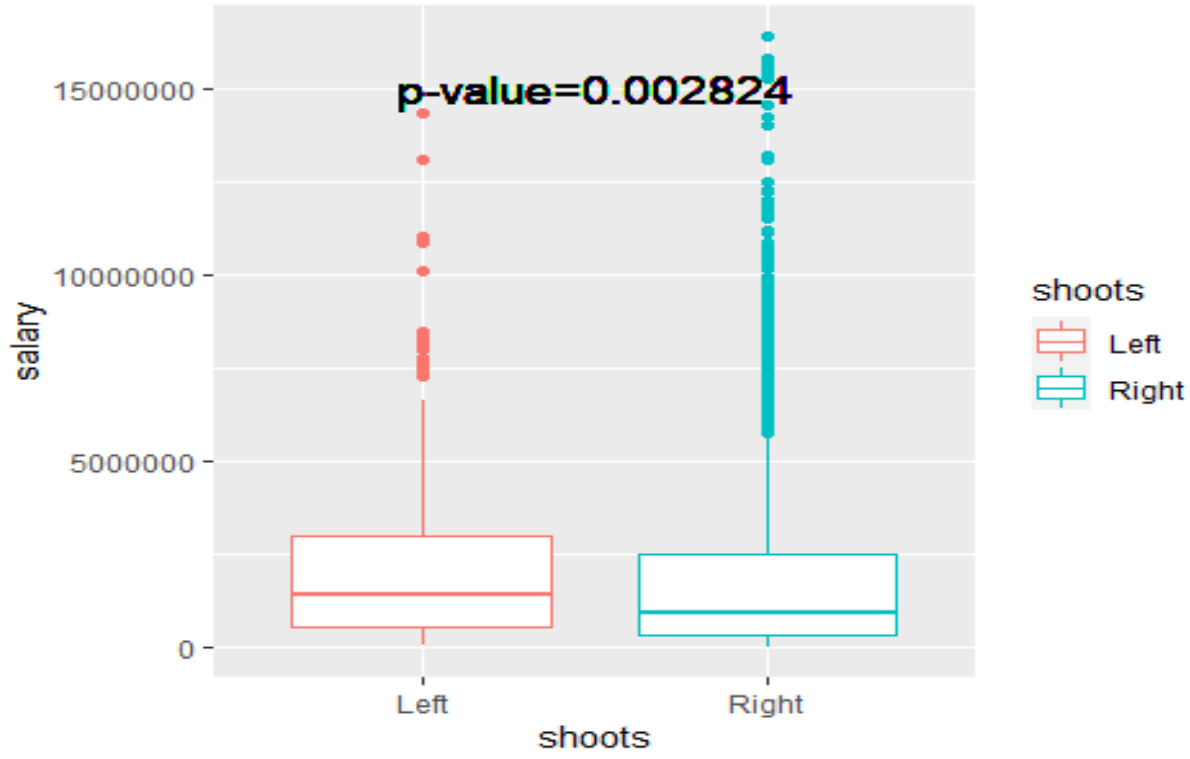
- wilcox test 비모수검정

```
wilcox.test(shoots_l$salary,shoots_r$salary,
            alternative = c("two.sided","less","greater"),
            mu = 0,
            conf.int =FALSE,
            conf.level = 0.95)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  shoots_l$salary and shoots_r$salary
## W = 223932, p-value = 0.002824
## alternative hypothesis: true location shift is not equal to 0
```

시각화

```
ggplot(shoots_all,aes(x=shoots,y=salary,col=shoots))+
  geom_boxplot()+
  geom_text(aes(x=1.5,y=15000000,label='p-value=0.002824'),color='black',size
=5)
```



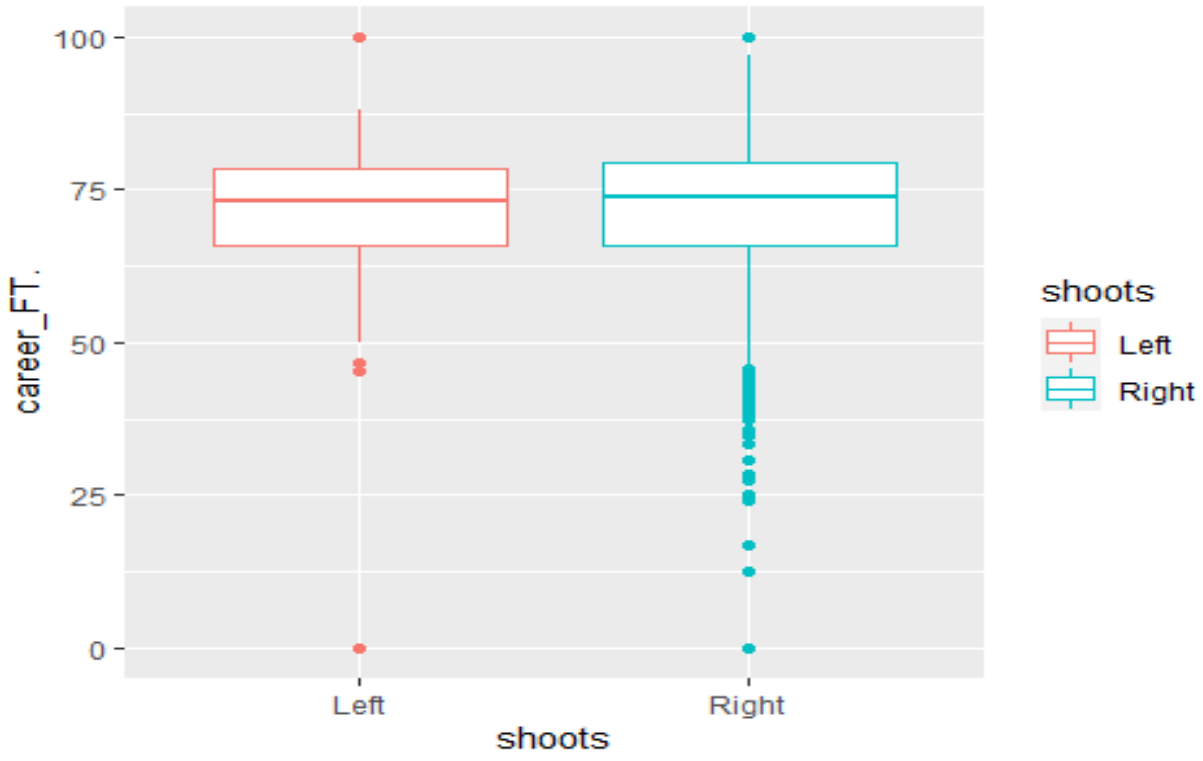
- 3.1.2 왼손 선수집단과 오른손 선수집단의 자유투성공율에는 차이가 있을 것이다.

  전처리

```
df2$career_FT. = as.numeric(df2$career_FT.)
shoots_R = subset(df2,shoots=='Right')
shoots_L = subset(df2,shoots=='Left')
df_shoots =rbind(shoots_R,shoots_L)
```

- Two sample t-test

```
ggplot(df_shoots,aes(x=shoots,y=career_FT.,col=shoots))+
  geom_boxplot()
```

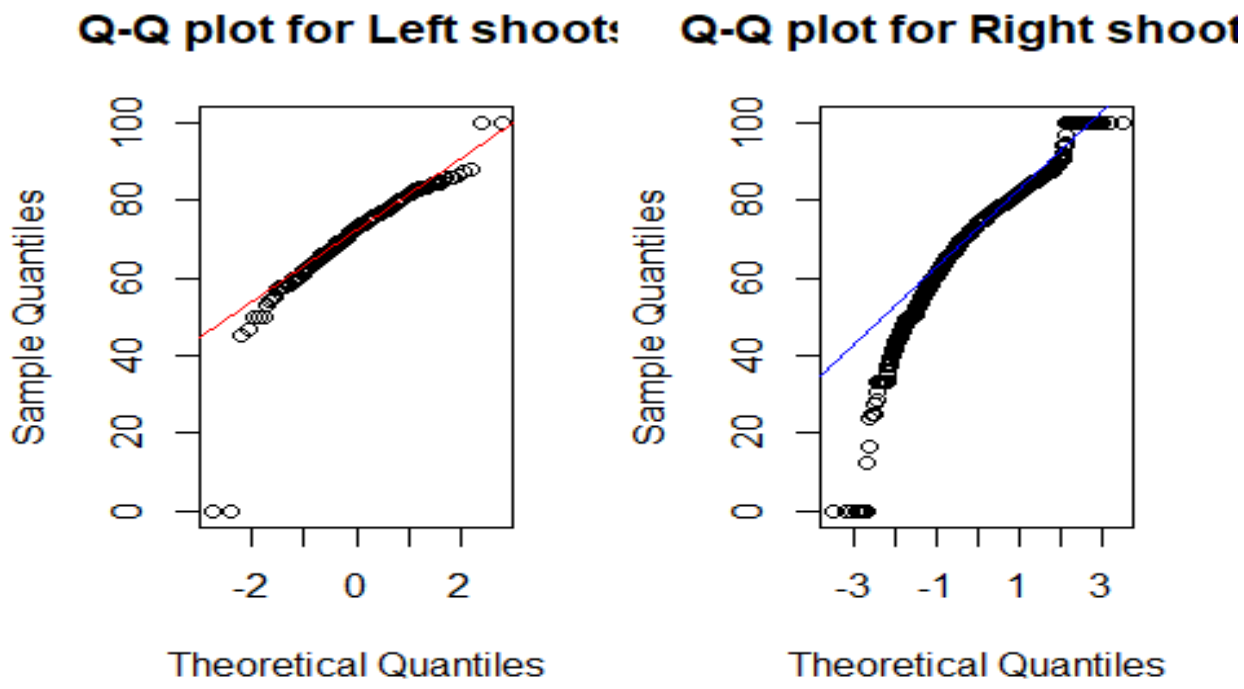


등분산 검정 var.test

```
var.test(career_FT.~shoots,data=df_shoots)

##
##  F test to compare two variances
##
## data:  career_FT. by shoots
## F = 0.97657, num df = 170, denom df = 2166, p-value = 0.8591
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7909625 1.2315379
## sample estimates:
## ratio of variances
##          0.9765714
```

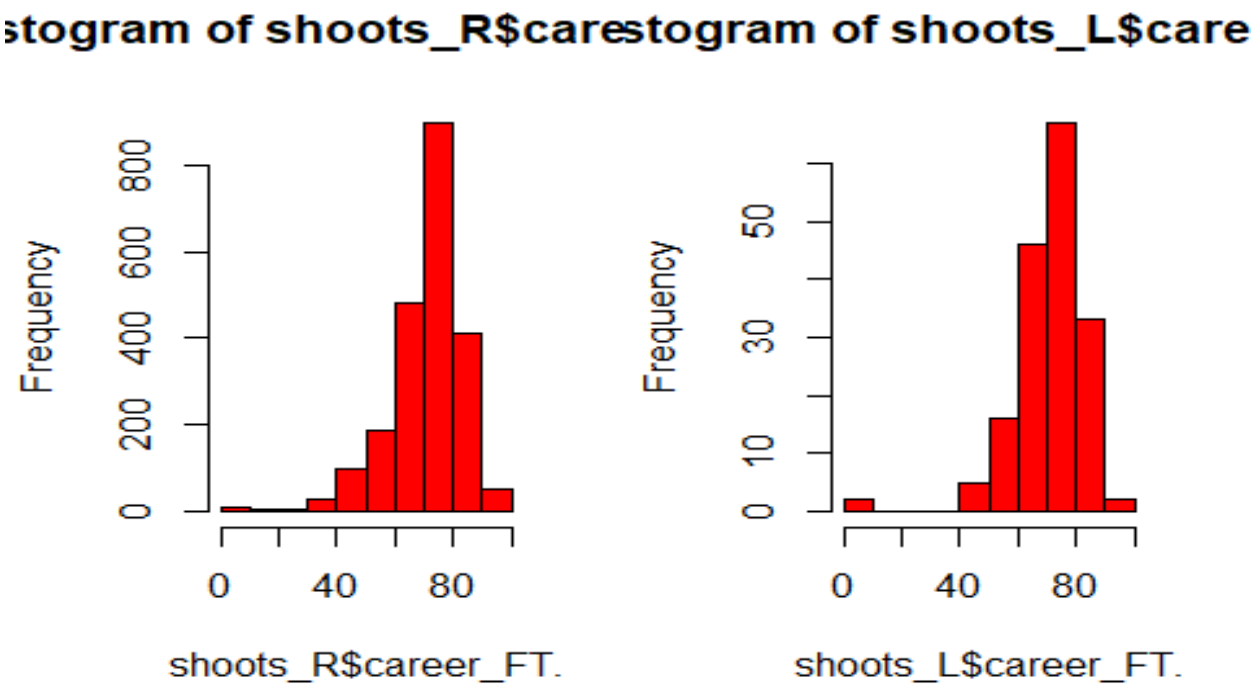
정규성 검정(1) qqplot

```
par(mfrow=c(1,2))
qqnorm(shoots_L$career_FT.,main="Q-Q plot for Left shoots");qqline(shoots_L$c
areer_FT.,col='red')
qqnorm(shoots_R$career_FT,main="Q-Q plot for Right shoots");qqline(shoots_R$c
areer_FT,col='blue')
```



정규성 검정(2) plot histogram

```
par(mfrow=c(1,2))
hist(shoots_R$career_FT.,breaks=10,col=2)
hist(shoots_L$career_FT.,breaks=10,col=2)
```



정규성 검정(3) shapiro test

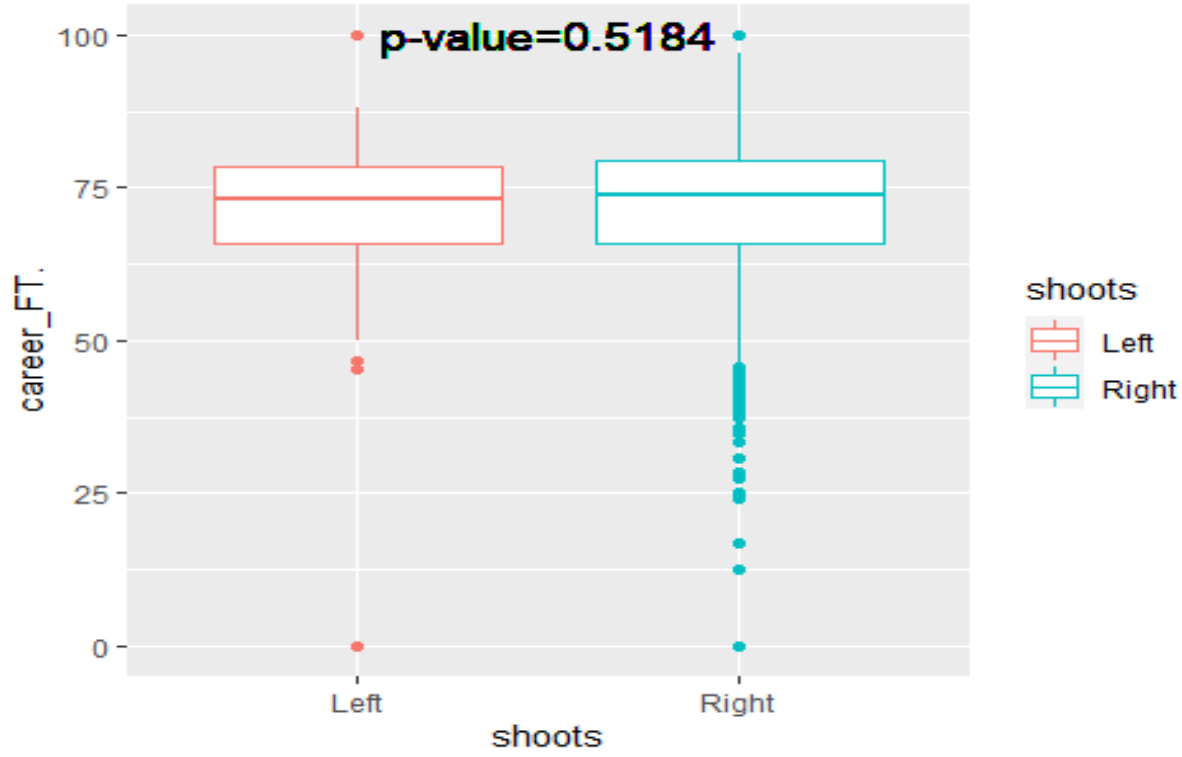
```
options(scipen=999)
shapiro.test(shoots_L$career_FT.)

##
##  Shapiro-Wilk normality test
##
## data:  shoots_L$career_FT.
## W = 0.83688, p-value = 0.000000000001522

shapiro.test(shoots_R$career_FT.)

##
##  Shapiro-Wilk normality test
##
## data:  shoots_R$career_FT.
## W = 0.92006, p-value < 0.00000000000000022
```

- Wilcox test 비모수검정

```
wilcox.test(shoots_L$career_FT.,shoots_R$career_FT.,
            alternative = c("two.sided","less","greater"),
            mu = 0,
            conf.int =FALSE,
            conf.level = 0.95)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  shoots_L$career_FT. and shoots_R$career_FT.
## W = 179789, p-value = 0.5184
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(df_shoots,aes(x=shoots,y=career_FT.,col=shoots))+
  geom_boxplot()+
  geom_text(aes(x=1.5,y=100,label='p-value=0.5184'),color='black',size=5)
```



3.1.3 왼손 선수집단과 오른손 선수집단의 드래프트 라운드에는 차이가 있을 것이다. 선수의 드래프트 라운드 중 1 부터 5 라운드까지 왼손 선수와 오른손 선수의 차이가 있는지 확인.

```
s= df2$shoots
r= df2$draft_round
sr = data.frame(s,r)
sr = sr[(sr$r=='1st round') |
          (sr$r=='2nd round')|
          (sr$r=='3rd round')|
          (sr$r=='4th round')|
          (sr$r=='5th round'),]
sr = sr[!(sr$s =='Left Right'),]
sr$s= factor(sr$s)
```

```
sr$r= factor(sr$r,levels=c("1st round","2nd round","3rd round", "4th round",
"5th round"))
```

교차분석 sjt.xtab

```
sjt.xtab(sr$r,sr$s,
         show.col.prc=T,
         show.exp=T,
         var.labels=c("round","Primary Hands"),
         encoding='EUC-KR')
```

round	*Primary Hands*		***Total***
	Left	Right	
1st round	90 78 66.7 %	984 996 57.1 %	1074 1074 57.8 %
2nd round	40 50 29.6 %	651 641 37.8 %	691 691 37.2 %
3rd round	2 4 1.5 %	51 49 3 %	53 53 2.9 %
4th round	2 2 1.5 %	31 31 1.8 %	33 33 1.8 %
5th round	1 1 0.7 %	6 6 0.3 %	7 7 0.4 %
Total	135 135 100 %	1723 1723 100 %	1858 1858 100 %

χ^2=5.758 · df=4 · Cramer's V=0.056 · Fisher's p=0.170

 # 결과해석 : 두 범주형 변수로 교차분석을 실시해본 결과 p-value 값이 0.178 로 대립가설이 기각될 수 있는 유의수준인 0.05 보다 큰값을 가지므로 대립가설은 기각되며 "선수의 주 손잡이에 따라 지명되는 드래프트 라운드의 분포의 차이가 없을것"이라는 귀무가설이 그대로 채택됩니다.

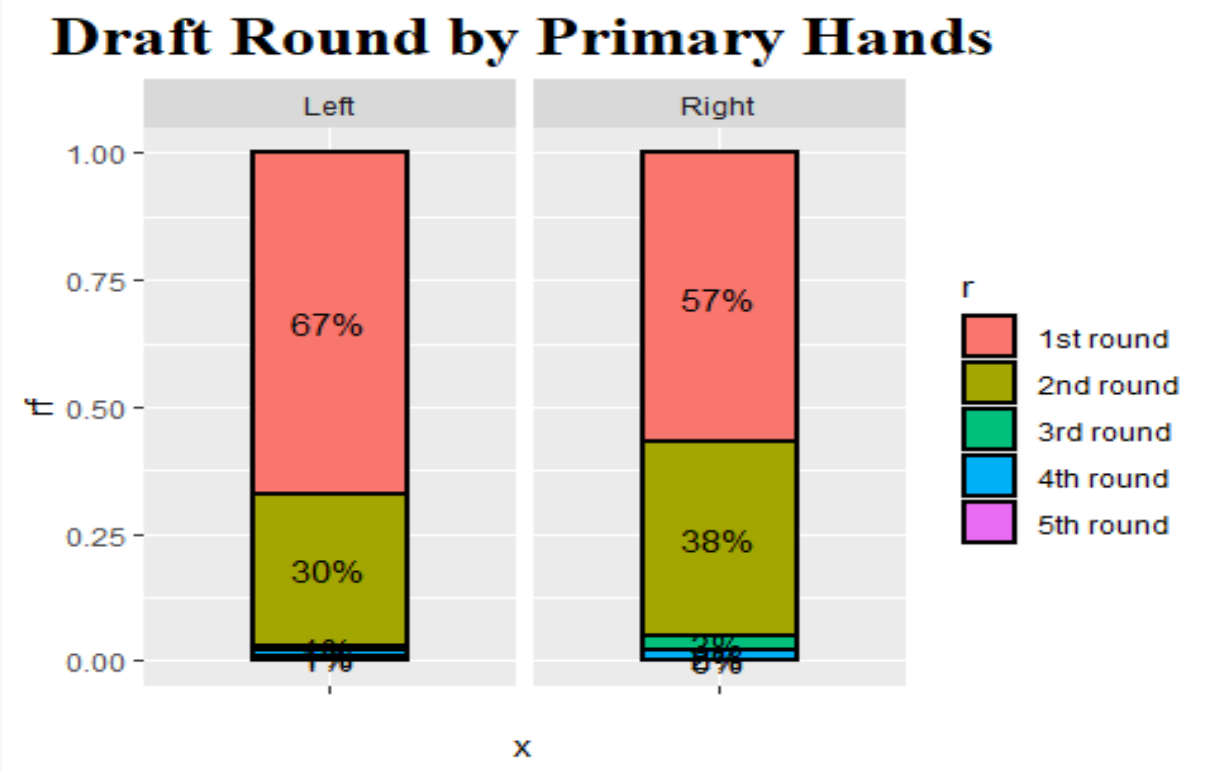
```
sr_table=table(sr)
sr_table = data.frame(sr_table)

sr_left = sr_table[sr_table$s=='Left',]
sr_right = sr_table[sr_table$s=='Right',]

sr_left
```

```
##      s          r Freq
## 1 Left 1st round   90
## 3 Left 2nd round   40
## 5 Left 3rd round    2
## 7 Left 4th round    2
## 9 Left 5th round    1
```

```
sr_left$rf = round(sr_left$Freq/sum(sr_left$Freq),2)
sr_right$rf = round(sr_right$Freq/sum(sr_right$Freq),2)

sr_c = rbind(sr_left,sr_right)
```

시각화

```
ggplot(sr_c,aes(x="",y= rf,fill=r))+
  geom_bar(width=0.5,stat= 'identity',color='black',size=1.0)+
  facet_grid(facets = .~s)+
  ggtitle("Draft Round by Primary Hands")+
  theme(plot.title = element_text(family='serif',face="bold",hjust=0.5,size=2
0,color='black'))+
  geom_text(aes(label =paste0(round(rf*100,1),"%")),
            position = position_stack(vjust = 0.5))
```



3.2 선수의 stat 에 따른 연봉차이 추론통계

- 3.2 선수의 stat 에 따른 연봉차이 추론통계
- 3.2.1 연봉 상위권 계층과 하위권 계층간의 Career_WS 값은 유의미한 차이를 보일것이다. 필요한 데이터만 담은 sub_df 생성

```
sub_df = data.frame(x_id = df2$X_id,career_WS = df2$career_WS,salary = df2$sa
lary)
```

연봉 범주화 (summary 활용)

```
summary(sub_df$salary)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     2706   349250   931784  1964587  2535453 16411903
```

상위 25% 연봉 선수 범주확인 (max ~ 3rd Qu)

```
summary(sub_df$salary)[6] # max = 16411903 ~
```

```
##     Max.
## 16411903
```

```
summary(sub_df$salary)[5] # 3rd qu = 2535453
```

```
## 3rd Qu.
## 2535453
```

하위 25% 연봉 선수 범주확인 (1st Qu ~ min)

```
summary(sub_df$salary)[2] # 3rd qu = 349249
```

```
##  1st Qu.
## 349249.8
```

```
summary(sub_df$salary)[1] # min = 2706
```

```
## Min.
## 2706
```

상위 25% 범주화

```
sub_df_top25 = sub_df[sub_df$salary <= 16411903 & sub_df$salary > 2535453,]
sub_df_top25$TB = 'salary_Top25'
```

하위 25% 범주화

```
sub_df_bot25 = sub_df[sub_df$salary <= 349249 & sub_df$salary > 2706,]
sub_df_bot25$TB = 'salary_Bottom25'
```

데이터프레임 재구성

```
TB_25_salary_df = rbind(sub_df_top25,sub_df_bot25)
TB_25_salary_df$TB = factor(TB_25_salary_df$TB)
```

승리 기여도 범주화

```
summary(TB_25_salary_df$career_WS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -2.70    0.20    8.60   23.05   33.88  234.60
```

```
#(-2.7 / 0.20 / 8.60 / 33.88 / 234.60 )
str(TB_25_salary_df$career_WS)
```

```
##  num [1:1202] 3.5 71.2 5 38.7 33.5 ...
```

```
Qu4 = TB_25_salary_df[TB_25_salary_df$career_WS <= 234.60 & TB_25_salary_df$c
areer_WS > 33.88,]
Qu4$ws_Qu = 'Qu4'

Qu3 = TB_25_salary_df[TB_25_salary_df$career_WS <= 33.88 & TB_25_salary_df$ca
reer_WS > 8.60,]
Qu3$ws_Qu = 'Qu3'

Qu2 = TB_25_salary_df[TB_25_salary_df$career_WS <= 8.60 & TB_25_salary_df$car
eer_WS > 0.20,]
Qu2$ws_Qu = 'Qu2'

Qu1 = TB_25_salary_df[TB_25_salary_df$career_WS <= 0.20 & TB_25_salary_df$car
eer_WS > -2.70,]
Qu1$ws_Qu = 'Qu1'

TB_25_salary_WS_df = rbind(Qu4,Qu3,Qu2,Qu1)
TB_25_salary_WS_df$ws_Qu= factor(TB_25_salary_WS_df$ws_Qu)
```

교차분석표

```
sjt.xtab(TB_25_salary_WS_df$ws_Qu,TB_25_salary_WS_df$TB,
         show.col.prc=T,
         show.exp=T,
         var.labels=c("승리기여도범주","연봉범주"),
         value.labels=list(levels(TB_25_salary_WS_df$TB),c('salary_Bottom25',
'salary_Top25')),encoding='EUC-KR')
## `var.row`.
```

승리기여도범주	연봉범주		Total
	salary_Bottom25	salary_Top25	
Qu1	312 160 52 %	8 160 1.3 %	320 320 26.6 %
Qu2	224 140 37.3 %	56 140 9.3 %	280 280 23.3 %
Qu3	55 150 9.2 %	245 150 40.8 %	300 300 25 %
Qu4	9 150 1.5 %	292 151 48.6 %	301 301 25.1 %
Total	600 600 100 %	601 601 100 %	1201 1201 100 %

χ^2=776.009 · df=3 · Cramer's V=0.804 · p=0.000

-
- *# 결과해석 : 카이제곱값은 776 이고 p-value 는 영가설을 기각하거나 채택할 수 있는 기준인 0.05 보다 낮기때문에 , 연봉 상/하위 25%에 해당하는 두개의 선수집단은 승리기여도 분포에 유의미한 차이를 보일것이란 대립가설을 채택합니다.*

- 3.2.2 연봉 상위권 계층과 하위권 계층간의 Career_PTS 값은 유의미한 차이를 보일것이다. 필요한 데이터만 담은 sub_df 생성

```
sub_df1 = data.frame(x_id = df2$X_id,career_PTS = df2$career_PTS,salary = df2
$salary)
```

연봉 범주화 (summary 활용)

```
summary(sub_df$salary)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     2706   349250   931784  1964587  2535453 16411903
```

상위 25% 연봉 선수 범주확인 (max ~ 3rd Qu)

```
summary(sub_df$salary)[6] # max = 16411903 ~
```

```
##     Max.
## 16411903
```

```
summary(sub_df$salary)[5] # 3rd qu = 2535453
```

```
## 3rd Qu.
## 2535453
```

하위 25% 연봉 선수 범주확인 (1st Qu ~ min)

```
summary(sub_df$salary)[2] # 3rd qu = 349249

##  1st Qu.
## 349249.8

summary(sub_df$salary)[1] # min = 2706

## Min.
## 2706
```

상위 25% 범주화

```
sub_df1_top25 = sub_df1[sub_df1$salary <= 16411903 & sub_df1$salary > 253545
3,]
sub_df1_top25$TB = 'salary_Top25'
```

하위 25% 범주화

```
sub_df1_bot25 = sub_df1[sub_df1$salary <= 349249 & sub_df1$salary > 2706,]
sub_df1_bot25$TB = 'salary_Bottom25'
```

데이터 프레임 재구성

```
TB_25_salary_df1 = rbind(sub_df1_top25,sub_df1_bot25)
TB_25_salary_df1$TB = factor(TB_25_salary_df1$TB)
```

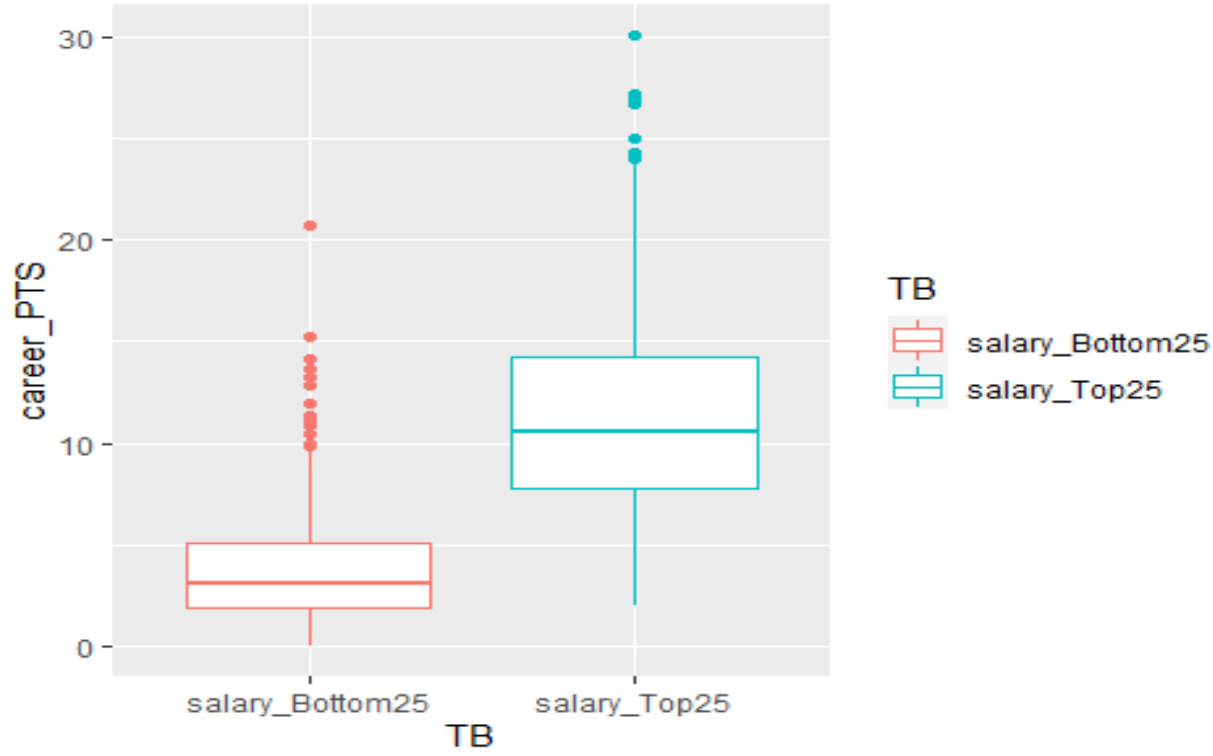
득점 포인트 범주화 (summary 활용)

```
summary(TB_25_salary_df1$career_PTS)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.100   6.400   7.525  10.900  30.100

#(0.000 / 3.100 / 6.400 / 10.900 / 30.100 )
Qu_4 = TB_25_salary_df1[TB_25_salary_df1$career_PTS <= 30.1 & TB_25_salary_df
1$career_PTS > 10.9,]
Qu_4$ws_Qu = 'Qu4'

Qu_3 = TB_25_salary_df1[TB_25_salary_df1$career_PTS <= 10.9 & TB_25_salary_df
1$career_PTS > 6.4,]
Qu_3$ws_Qu = 'Qu3'

Qu_2 = TB_25_salary_df1[TB_25_salary_df1$career_PTS <= 6.4 & TB_25_salary_df1
$career_PTS > 3.1,]
```

```
Qu_2$ws_Qu = 'Qu2'

Qu_1 = TB_25_salary_df1[TB_25_salary_df1$career_PTS <= 3.1 & TB_25_salary_df1
$career_PTS > 0,]
Qu_1$ws_Qu = 'Qu1'

TB_25_salary_WS_df1 = rbind(Qu_4,Qu_3,Qu_2,Qu_1)
TB_25_salary_WS_df1$ws_Qu= factor(TB_25_salary_WS_df1$ws_Qu)
```

- Two sample t-test

```
ggplot(TB_25_salary_df1,aes(x=TB,y=career_PTS,col=TB))+
  geom_boxplot()
```



등분산 검정

```
var.test(career_PTS~TB,data=TB_25_salary_df1)

##
##  F test to compare two variances
##
## data:  career_PTS by TB
## F = 0.31469, num df = 600, denom df = 600, p-value <
## 0.00000000000000022
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2681154 0.3693640
```

```
## sample estimates:
## ratio of variances
##          0.3146938
```

Welch Two Sample t-test

```
t.test(career_PTS~TB,TB_25_salary_df1,var.equal=F)

##
##  Welch Two Sample t-test
##
## data:  career_PTS by TB
## t = -33.521, df = 943.6, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.922005 -7.045716
## sample estimates:
## mean in group salary_Bottom25    mean in group salary_Top25
##                      3.783195                      11.267055
```

IV. 결론

4.1 가설검증 결과

1. 대립가설: **"왼손 선수집단과 오른손 선수집단의 평균연봉에는 차이가 있을 것이다."** 데이터의 정규성검사결과 정규성이 없다고 판단되었고, 이에 따라 wilcox test 를 실시한 결과 귀무가설이 기각되어 "왼손 선수집단과 오른손 선수집단의 평균연봉에는 차이가 있을 것이다"라는 대립가설이 채택되었습니다.

2. 대립가설: **" 왼손 선수집단과 오른손 선수집단의 자유투성공률에는 차이가 있을 것이다."** 이 경우도 마찬가지로 정규성이 없다고 판단되어 wilcox test 를 실시한 결과, "좌투수/우투수 간 자유투 성공률 평균에 유의미한 차이는 존재하지 않는다"는 귀무가설은 기각되지 않으므로 해당 대립가설은 기각되었습니다.

3. 대립가설: **"왼손 선수집단과 오른손 선수집단의 드래프트 라운드에는 차이가 있을 것이다. "** 두 범주형 변수로 교차분석을 실시해본 결과"선수의 주

손잡이에 따라 선발되는 드래프트 라운드의 분포의 차이가 없을것"이라는 귀무가설이 그대로 채택되며 따라서 해당 대립가설은 기각되었습니다.

4. 대립가설: **"연봉 상위권 계층과 하위권 계층 간의 승리기여도값은 유의미한 차이를 보일것이다."** 선수의 연봉을 기준으로 연봉 상위 25% /하위 25%값에 범주화 시켜 승리기여도의 범주와 교차분석한 결과 승리기여도 값은 유의미한 차이를 보일 것이란 대립가설이 채택되고, 귀무가설은 기각되었습니다.

5. 대립가설: **"연봉 상위권 계층과 하위권 계층간의 평균득점값은 유의미한 차이를 보일것이다. "**해당 두 변수를 범주화하여 박스플롯으로 시각화해본 결과 연봉 상위 25% 선수집단이 하위집단에 비해 유의미하게 높은 득점평균을 보이고 있었으며, 해당 데이터가 등분산을 만족시키지 못했기 때문에 Welch's t-검정을 실시한 결과, 박스플롯에서 유추한 결과와 동일하게 대립가설이 채택되었습니다.

4.2 결론

- 각 대립가설에 해당되는 변수의 유형에 따라 교차분석, T 검정(welch t 검정/wilcox 검정)을 실시해본 결과, "좌투수/우투수 간 자유투 성공률 평균에 유의미한 차이는 존재할 것"이란 연구가설과 "왼손 선수집단과 오른손 선수집단의 드래프트 라운드에는 차이가 있을 것"이란 연구가설 그리고 "선수의 주 손잡이에 따라 선발되는 드래프트 라운드의 분포의 차이가 존재할 것"이란 연구가설은 기각되었습니다.

- 반면에 "왼손 선수집단과 오른손 선수집단의 평균연봉에는 차이가 존재할 것"이란 연구가설이 채택되어 선수의 주 손이 선수의 연봉책정에 유의미한 영향력을 가진다는 것을 알아낼 수 있었고. "연봉 상위권 계층과 하위권 계층 간의 승리기여도 값은 유의미한 차이를 보일 것"이란 연구가설이 채택되어 연봉의 계층에 따라 승리기여도에 유의미한 영향을 줄 수 있다는 사실도 알아낼 수 있었습니다.

- 평소 관심있게 지켜보던 NBA 선수셋을 데이터로 기술통계/추론통계를 실시하여 드래프트 라운드 변수가 가지는 유의미한 분석결과를 필두로 각종 변수가 보여주는 다양하고 흥미로운 결과물을 도출하였습니다.

- 해당 보고서의 유의미한 분석결과들은 Moneyball 로 유명한 MLB 에서의 데이터분석의 활용사례와 같이 NBA 에서 또한 리그에서 수집되는 정량적인 데이터로부터 충분히 유의미한 insight 를 도출해내고 활용할 수 있음을 시사하는 바입니다. 후속 연구에서는 해당강의 하반기의 교육과정에 포함된 분산분석 이하의 분석기법을 도입하여 추가적인 분석을 실시해보고자 합니다.

4.3 한계점

- 1985 년부터 2018 년까지 선수들의 연봉정보 등 전반적인 인적사항 데이터를 활용하여 분석해보았습니다. 하지만 2020 년인 현재 시점에서, 서로 다른 시점의 금액을 비교하였고 시간이 지남에 따라 발생한 인플레이션 효과가 있었을 것입니다. 저희 팀은 선수의 커리어 기간 동안의 총 평균임금을 책정한 데이터로 분석할 수밖에 없었습니다. 현재 물가수준과 과거의 물가수준을 반영한 인플레이션 보정 공식을 통하여 계산했다면 현재의 화폐가치로 환산하여 훨씬 정확도가 높았을 것이라는 아쉬움이 남습니다.