“SDS_HW03”

2017100057 / 이영노

November 17, 2022

Question 1

1.1

```
take_sample=function(p,N){
  x=sample(c(0,1),size=N,replace=TRUE,prob=c(1-p,p))
  mean(x)
}
```

1.2

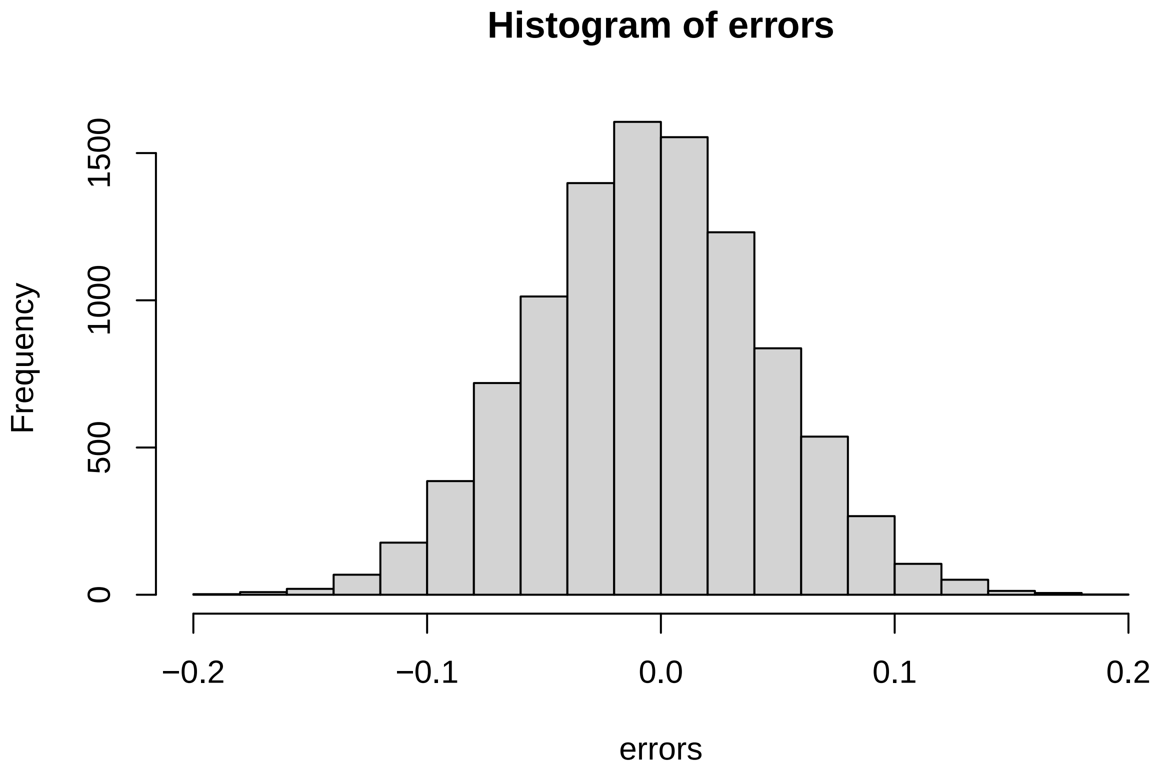
```
p=.45; N=100; nsim=10000;
set.seed(2022)
errors=replicate(nsim,{
  x=take_sample(p,N)
  error=p-mean(x)
})
```

1.3

```
mean(errors)
```

```
## [1] 1.1e-05
```

```
hist(errors)
```

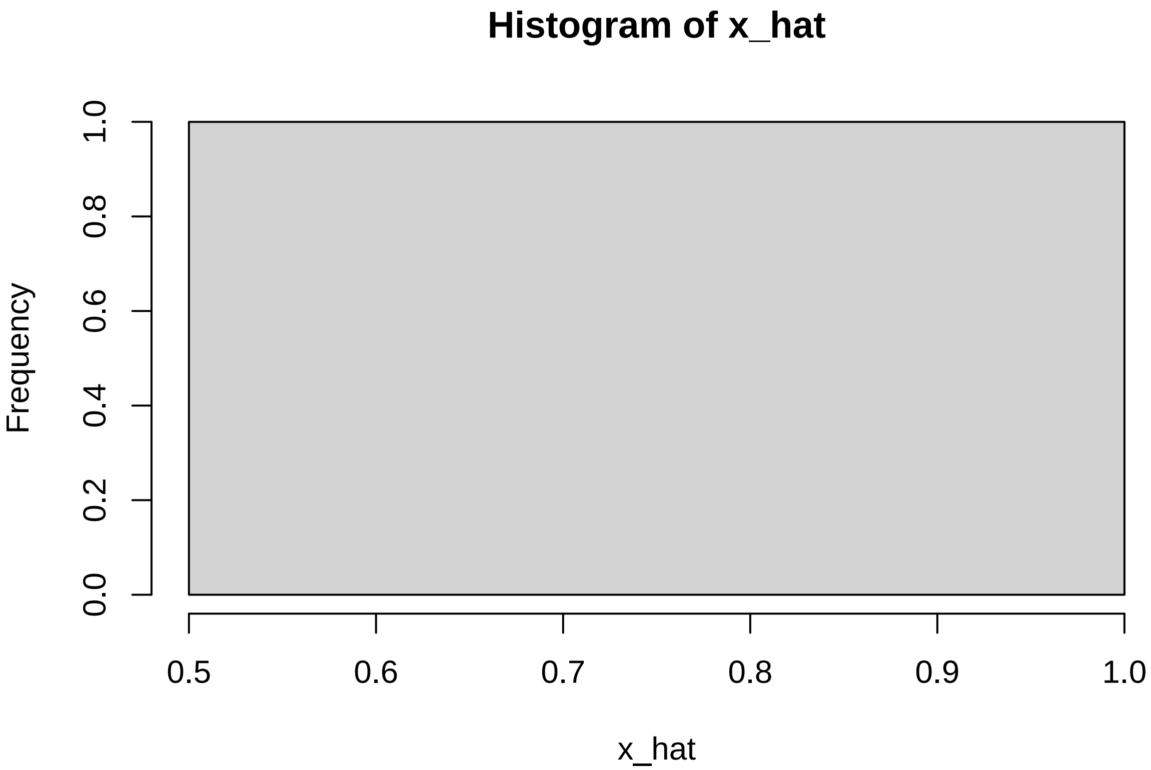


By looking at the histogram, error is almost symmetrically distributed around 0.

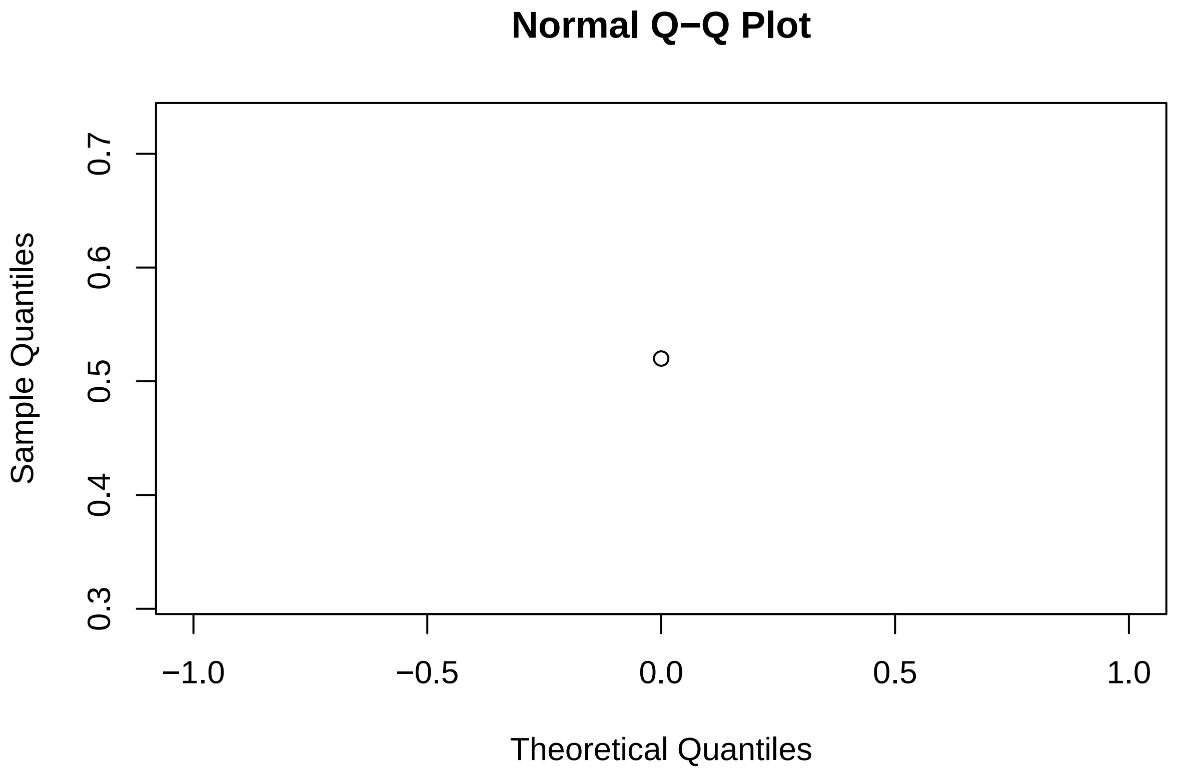
So, (c) best describes distribution.

1.9

```
p=.45; N=100; x_hat=take_sample(p,N) # using code from question 1.1
hist(x_hat)
```


Histogram of x_hat
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
0.5
0.6
0.7
0.8
0.9
1.0
x_hat

```
qqnorm(x_hat)
```



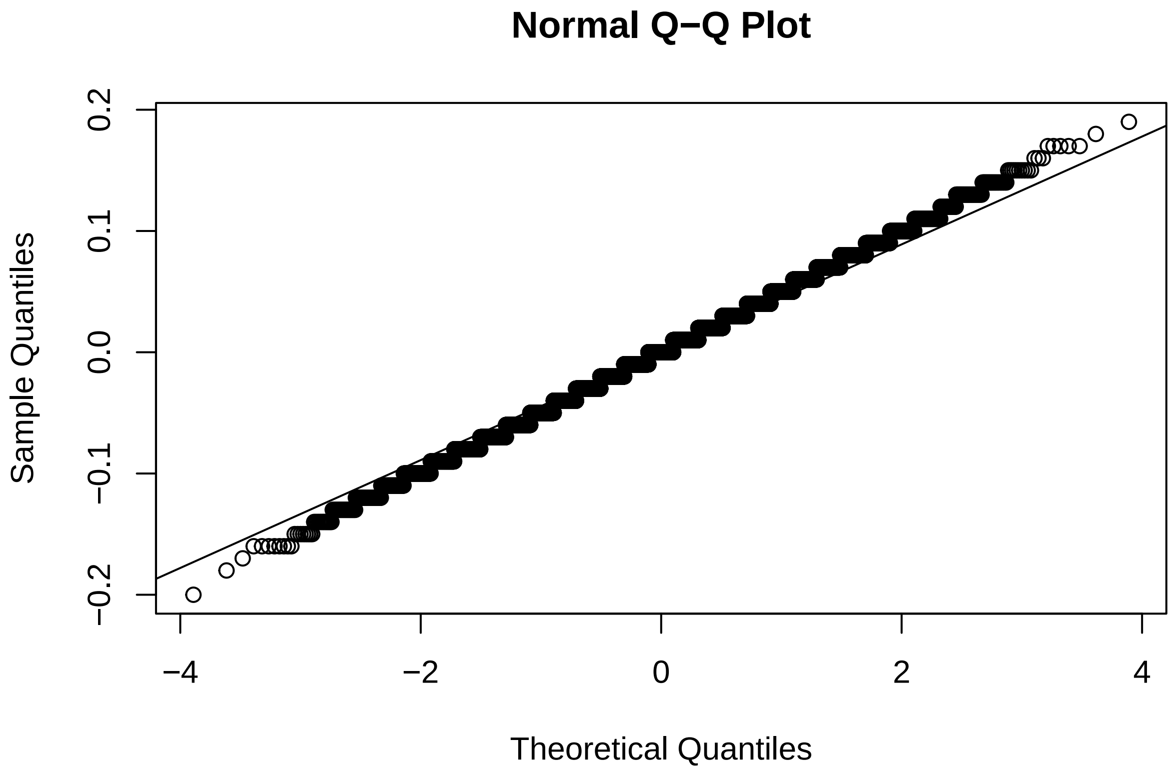
When sample size is large, CLT tells that distribution of sample proportion approximately follows normal distribution which have expected value of `p` and SE of `sqrt(p*(1-p)/N)`. Thus, (b) is correct.

1.10

Since sample proportion (p_{hat}) approximately follows Normal distribution with N(p,sqrt(p(1-p)/N)), ($p_{hat} - p$) follows N(0,sqrt(p(1-p)/N)). Thus, (b) is correct.

1.11

```
p=0.45; N=100; nsim=10000; set.seed(2022);
qqnorm(errors)
qqline(errors)
```



Errors follow Normal distribution adaequately.

1.12

```
p=.45; se=sqrt(p*(1-p)/N)
1-pnorm(0.5,p,se)
```

```
## [1] 0.1574393
```

Probability that $(\bar{X} > 0.5)$ is 0.1574393.

Question 2

SETUP

```
library(dslabs)
```

```
## Warning: 패키지 'dslabs'는 R 버전 4.1.3에서 작성되었습니다
```

```
data("polls_us_election_2016")
library(tidyverse)
```

```
## Warning: 패키지 'tidyverse'는 R 버전 4.1.3에서 작성되었습니다
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.7      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.2
```

```
## Warning: 패키지 'ggplot2'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'tibble'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'tidyr'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'readr'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'purrr'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'dplyr'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'stringr'는 R 버전 4.1.3에서 작성되었습니다
```

```
## Warning: 패키지 'forcats'는 R 버전 4.1.3에서 작성되었습니다
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
polls <- polls_us_election_2016 |>
  filter(enddate >= "2016-10-31" & state == "U.S.")

N=polls$samplesize[1]
x_hat=polls$rawpoll_clinton[1]/100
```

2.1

x_hat is sample proportion

```r
se=sqrt(x_hat*(1-x_hat)/N)
moe=pnorm(0.975)*se
CI=x_hat+c(-1,1)*moe
print(CI)
```

```
## [1] 0.4611527 0.4788473
```

95% Confidence Interval is as above.

2.2

```r
poll=polls%>%mutate(x_hat=polls$rawpoll_clinton/100,se_hat=sqrt(x_hat*(1-x_hat)/samplesize),
               lower=x_hat-pnorm(0.975)*se_hat,upper=x_hat+pnorm(.975)*se_hat)%>%
  select(pollster,enddate,x_hat,lower,upper)
poll%>%head(5)
```

```
##                   pollster    enddate  x_hat     lower     upper
## 1 ABC News/Washington Post 2016-11-06 0.4700 0.4611527 0.4788473
## 2  Google Consumer Surveys 2016-11-07 0.3803 0.3778127 0.3827873
## 3                    Ipsos 2016-11-06 0.4200 0.4112012 0.4287988
## 4                   YouGov 2016-11-07 0.4500 0.4431476 0.4568524
## 5         Gravis Marketing 2016-11-06 0.4700 0.4667684 0.4732316
```

2.3

```
hit_poll=poll%>%mutate(hit=lower<=0.482 & upper>=0.482)
hit_poll%>%head(6)
```

```
##                                                     pollster    enddate  x_hat
## 1                                   ABC News/Washington Post 2016-11-06 0.4700
## 2                                    Google Consumer Surveys 2016-11-07 0.3803
## 3                                                      Ipsos 2016-11-06 0.4200
## 4                                                     YouGov 2016-11-07 0.4500
## 5                                           Gravis Marketing 2016-11-06 0.4700
## 6 Fox News/Anderson Robbins Research/Shaw & Company Research 2016-11-06 0.4800
##       lower     upper   hit
## 1 0.4611527 0.4788473 FALSE
## 2 0.3778127 0.3827873 FALSE
## 3 0.4112012 0.4287988 FALSE
## 4 0.4431476 0.4568524 FALSE
## 5 0.4667684 0.4732316 FALSE
## 6 0.4684045 0.4915955  TRUE
```

2.4

```
hit_poll$hit%>%mean()
```

```
## [1] 0.1857143
```

CI includes p with probability of 0.1857143

2.5

It is 95% confidence interval, so if it is correctly constructed, it should include p with probability of 0.95.

2.6

```
polls <- polls_us_election_2016 |>
  filter(enddate >= "2016-10-31" & state == "U.S.")  |>
  mutate(d_hat = rawpoll_clinton / 100 - rawpoll_trump / 100)
N=polls$samplesize[1]
d_hat=polls$d_hat[1] # d_hat for the first pollster
x_hat=(d_hat+1)/2 # x_hat is Clinton's poll sample proportion.

se=2*sqrt(x_hat*(1-x_hat)/N)
d_hat+c(-1,1)*pnorm(.975)*se
```

```
## [1] 0.02228763 0.05771237
```

95% Confidence Interval is as above.

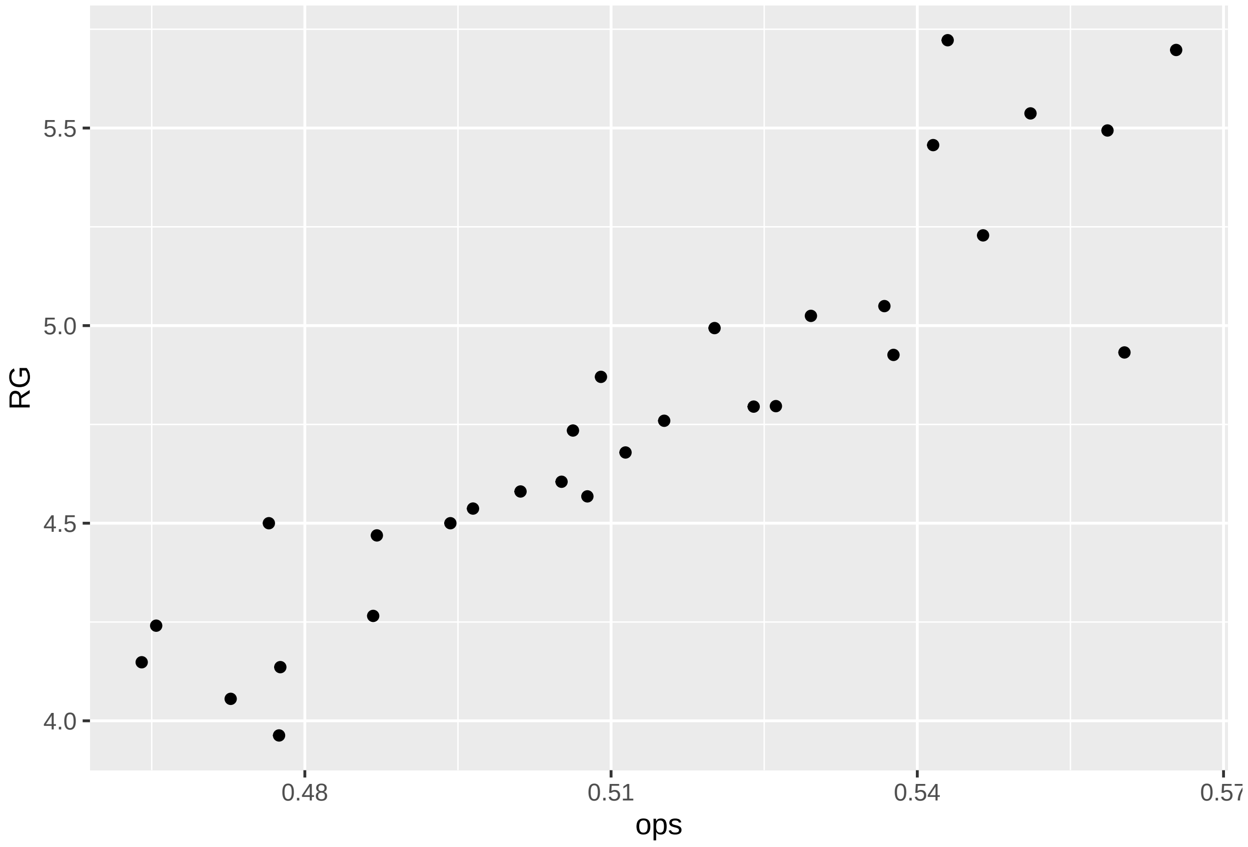
Question 3

3.1

```
library(Lahman)
```

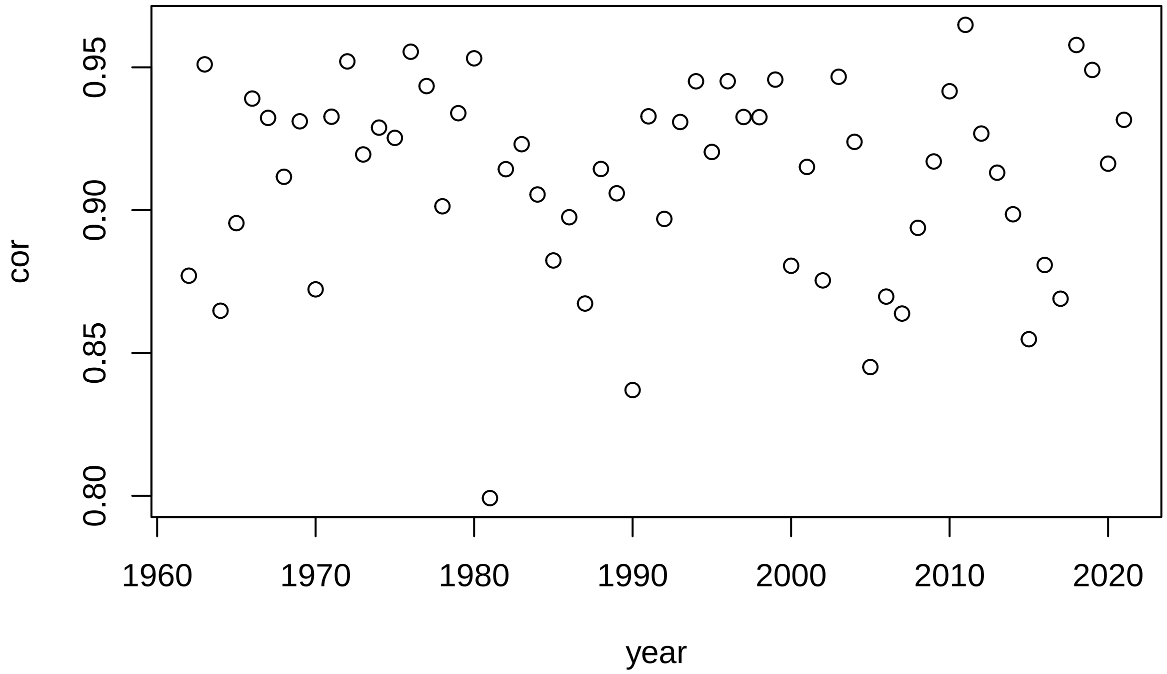
```
## Warning: 패키지 'Lahman'는 R 버전 4.1.3에서 작성되었습니다
```

```
data('Teams')
teams=Teams%>%mutate(pa=AB+BB, singles=(H - X2B - X3B - HR),doubles=X2B,
                     triples=X3B)%>%mutate(ops=BB/pa+(singles+2*doubles+3*triples+4*HR)/AB,
                                           RG=R/G)
teams%>%filter(yearID %in% 2001)%>%ggplot(aes(ops,RG))+
  geom_point()
```



3.2

```
df=teams%>%filter(yearID>=1962)%>%group_by(yearID)%>%summarize(cor=cor(RG,ops))
year=df%>%pull(yearID)
cor=df%>%pull(cor)
plot(year,cor)
```

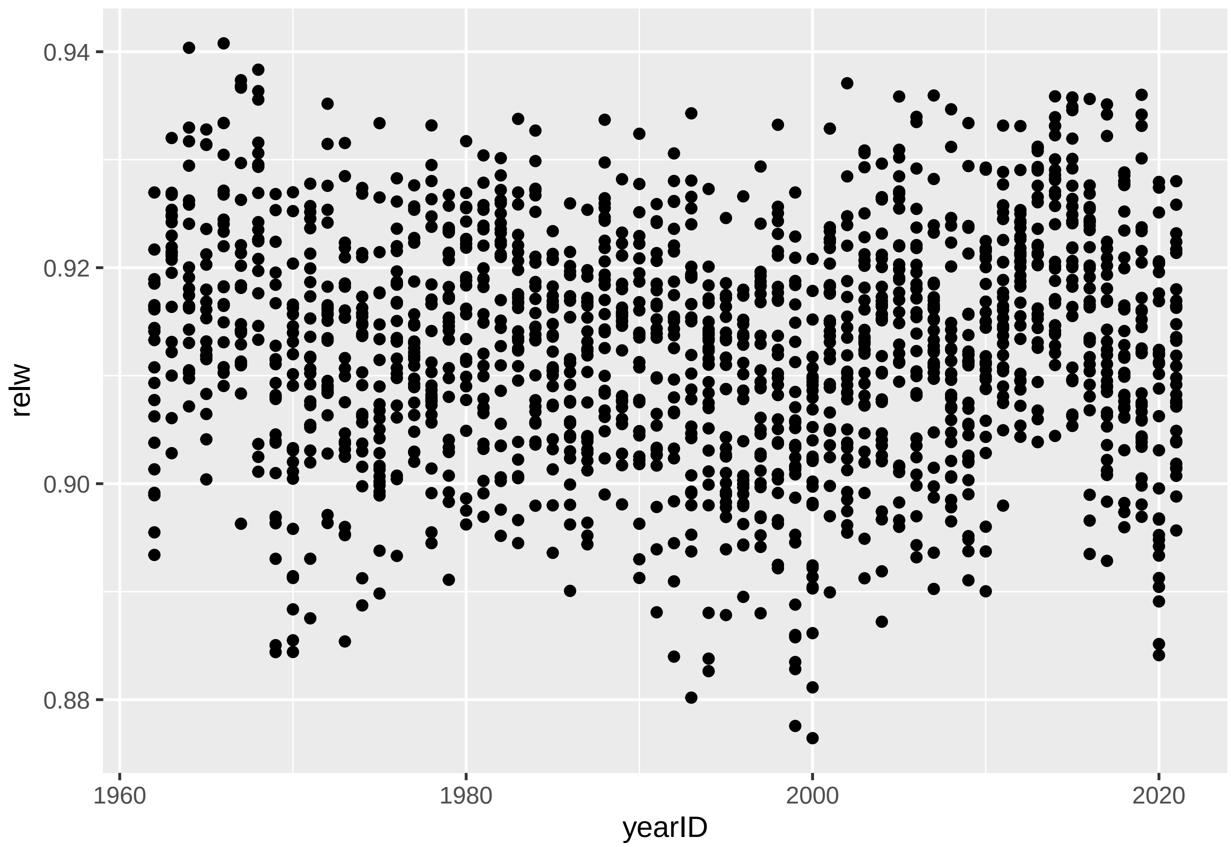


3.3

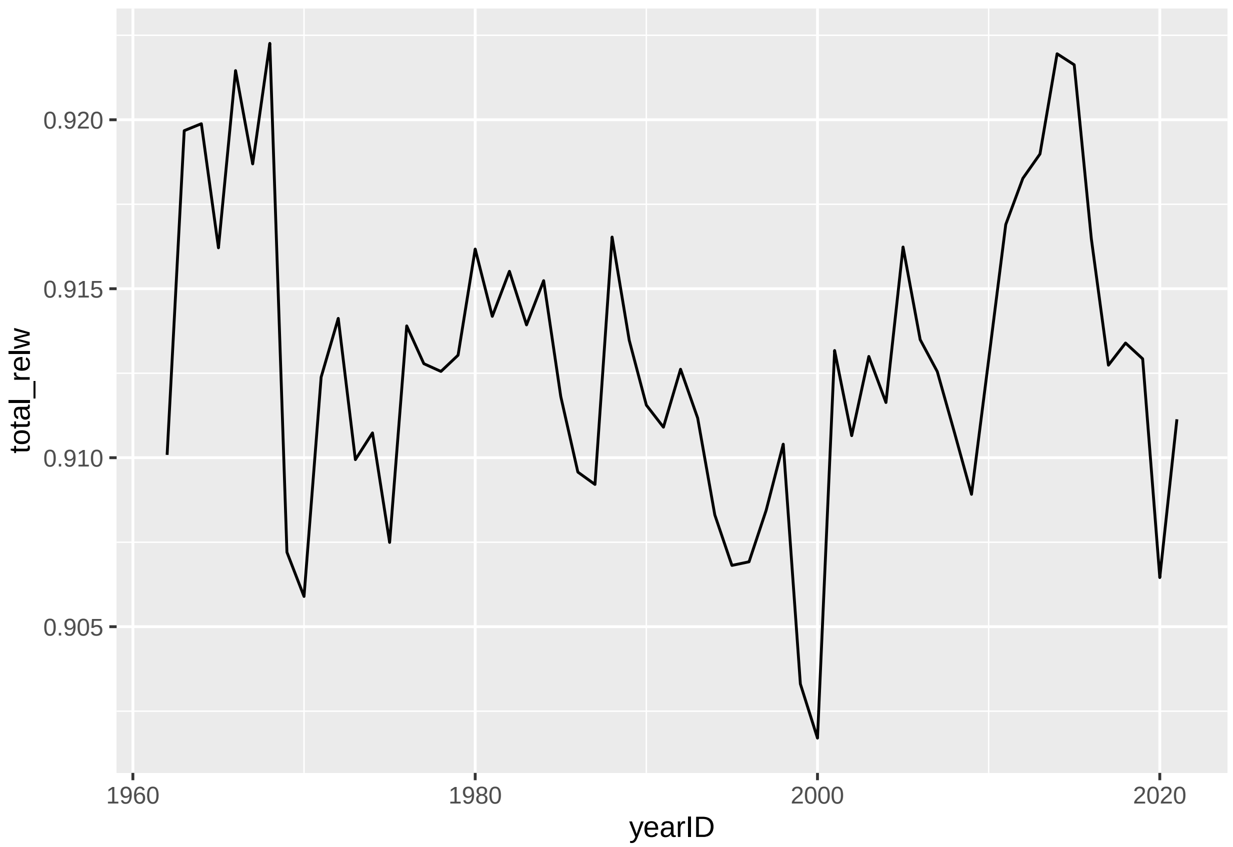
Weight for `BB` is 1/PA, and weight for `singles` is `1/AB`, so relative weight for BB to singles is AB/PA.

3.4

```
teams%>%filter(yearID>=1962)%>%group_by(yearID)%>%mutate(relw=AB/pa)%>%
  ggplot(aes(yearID,relw))+
  geom_point()
```



```
teams%>%filter(yearID>=1962)%>%group_by(yearID)%>%
  summarize(total_relw=sum(AB)/sum(pa))%>%
  ggplot(aes(yearID,total_relw))+
  geom_line()
```



```
teams%>%filter(yearID>=1962)%>%group_by(yearID)%>%
  summarize(total_relw=sum(AB)/sum(pa))%>%summarize(mean_relw=mean(total_relw))%>%
  pull(mean_relw)
```

```
## [1] 0.9128372
```

overall average is as above.

3.5

```
teams%>%filter(yearID>=1962)%>%mutate(singlesG=singles/G,doublesG=doubles/G,
                                      triplesG=triples/G,HRG=HR/G) %>%
  group_by(yearID,teamID)%>%
  lm(ops ~ singlesG + doublesG +triplesG + HRG, data= .) %>% .$coef
```

```
## (Intercept)    singlesG    doublesG    triplesG         HRG
##  0.16260422  0.02025322  0.04717280  0.06897431  0.11925776
```

```
mean_BBG=teams%>%mutate(BBG=BB/G)%>%filter(!is.na(BBG))%>%pull(BBG)%>%mean()
0.16260422/mean_BBG # different from 0.91
```

```
## [1] 0.05256687
```

as seen above coefficients, original coefficients doesn't fit well to the obtained regression.

reported coefficients are as above. singles per game coefficient is 0.02

coefficients are relatively almost 2,3,6 compared to coefficient of singlesG, therefore, original coefficients fit quite well.