# P80 Exercises

# Q1; Answer: 0.3007746

# Q2; Answer: 6.8%

# Q3; Answer: A)

# Q4; Answer: A)

# Q5; Answer: TRUE

# Q6; Answer: TRUE

# Q7; Answer: D)

# 以下、解答を導き出す根拠として使用したRソースコードを記載する

# Q1

set.seed(1);

dat<-rnorm(5,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数を5個発生させる

dat

mean(dat)

sd(dat) # 不偏分散による標本標準偏差

sqrt(5)\*mean(dat)/sd(dat) # Answer: 0.3007746

# Q2

set.seed(1);

B=1000 # 試行回数

n=5 # 1試行あたりの乱数の発生個数

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

hist(montecarlo)

sum(montecarlo>2)\*100/B # Answer: 6.8%

# Q3

# nを増やしてシミュレーションを繰り返していく

set.seed(1);

B=1000 # 試行回数

n=5 # 1試行あたりの乱数の発生個数(5個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

sum(montecarlo>2)/B # シミュレーション結果

1-pt(2,df=n-1) # 理論値

set.seed(1);

B=1000 # 試行回数

n=10 # 1試行あたりの乱数の発生個数(10個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

sum(montecarlo>2)/B # シミュレーション結果

1-pt(2,df=n-1) # 理論値

set.seed(1);

B=1000 # 試行回数

n=100 # 1試行あたりの乱数の発生個数(100個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

sum(montecarlo>2)/B # シミュレーション結果

1-pt(2,df=n-1) # 理論値

set.seed(1);

B=1000 # 試行回数

n=1000 # 1試行あたりの乱数の発生個数(1000個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

sum(montecarlo>2)/B # シミュレーション結果

1-pt(2,df=n-1) # 理論値

set.seed(1);

B=1000 # 試行回数

n=10000 # 1試行あたりの乱数の発生個数(10000個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

sum(montecarlo>2)/B # シミュレーション結果

1-pt(2,df=n-1) # 理論値

# Answer: A)

# Q4

# nを増やしてシミュレーションを繰り返していく

B=1000 # 試行回数

n=5 # 1試行あたりの乱数の発生個数(5個のとき)

montecarlo1 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo1 [i] <- sqrt(n)\*mean(dat1)/sd(dat1)}

montecarlo2 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat2<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo2 [i] <- sqrt(n)\*mean(dat2)/sd(dat2)}

DF<-2\*n-2 # 自由度

DF

res <- t.test(dat1, dat2, var.equal=TRUE)

res

B=1000 # 試行回数

n=10 # 1試行あたりの乱数の発生個数(10個のとき)

montecarlo1 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo1 [i] <- sqrt(n)\*mean(dat1)/sd(dat1)}

montecarlo2 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat2<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo2 [i] <- sqrt(n)\*mean(dat2)/sd(dat2)}

DF<-2\*n-2 # 自由度

DF

res <- t.test(dat1, dat2, var.equal=TRUE)

res

B=1000 # 試行回数

n=100 # 1試行あたりの乱数の発生個数(100個のとき)

montecarlo1 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo1 [i] <- sqrt(n)\*mean(dat1)/sd(dat1)}

montecarlo2 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat2<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo2 [i] <- sqrt(n)\*mean(dat2)/sd(dat2)}

DF<-2\*n-2 # 自由度

DF

res <- t.test(dat1, dat2, var.equal=TRUE)

res

B=1000 # 試行回数

n=1000 # 1試行あたりの乱数の発生個数(1000個のとき)

montecarlo1 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo1 [i] <- sqrt(n)\*mean(dat1)/sd(dat1)}

montecarlo2 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat2<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo2 [i] <- sqrt(n)\*mean(dat2)/sd(dat2)}

DF<-2\*n-2 # 自由度

DF

res <- t.test(dat1, dat2, var.equal=TRUE)

res

B=1000 # 試行回数

n=10000 # 1試行あたりの乱数の発生個数(10000個のとき)

montecarlo1 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo1 [i] <- sqrt(n)\*mean(dat1)/sd(dat1)}

montecarlo2 <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat2<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo2 [i] <- sqrt(n)\*mean(dat2)/sd(dat2)}

DF<-2\*n-2 # 自由度

DF

res <- t.test(dat1, dat2, var.equal=TRUE)

res

# Answer: A)

#Q5

X=rbinom(n=15,size=1,prob=0.5)

tstat <- sqrt(15)\*mean(X) / sd(X)

tstat

mean(X)

sd(X)

res <- t.test(X, mu=0.5)

res

# シミュレーションを繰り返すと、自由度14となり、概ねp>0.05の値が出る

# Answer: TRUE

#Q6

X=rbinom(n=500,size=1,prob=0.5)

tstat <- sqrt(500)\*mean(X) / sd(X)

tstat

mean(X)

sd(X)

res <- t.test(X, mu=0.5)

res

# シミュレーションを繰り返すと、自由度499となり、ほぼ確実にp>0.05の値が出る

# Answer: TRUE

# Q7

# nを増やしてシミュレーションを繰り返していく

B=1000 # 試行回数

n=5 # 1試行あたりの乱数の発生個数(5個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

mean(dat)

SE<- 1/sqrt(n)

SE

sd(dat)

B=1000 # 試行回数

n=10 # 1試行あたりの乱数の発生個数(10個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

mean(dat)

SE<- 1/sqrt(n)

SE

sd(dat)

B=1000 # 試行回数

n=100 # 1試行あたりの乱数の発生個数(100個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

mean(dat)

SE<- 1/sqrt(n)

SE

sd(dat)

B=1000 # 試行回数

n=1000 # 1試行あたりの乱数の発生個数(1000個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

mean(dat)

SE<- 1/sqrt(n)

SE

sd(dat)

B=1000 # 試行回数

n=10000 # 1試行あたりの乱数の発生個数(10000個のとき)

montecarlo <- rep(0, B) # 値の箱を作る

for(i in 1:B){ # 試行をB回繰り返す

dat1<-rnorm(n,mean=0,sd=1) # 標準正規分布N～(0,1)従う乱数をn個発生させる

montecarlo [i] <- sqrt(n)\*mean(dat)/sd(dat)}

mean(dat)

SE<- 1/sqrt(n)

SE

sd(dat)

# 試行回数を増やしても常にSE > sdとなる

# Answer: D)