Q2

I analyze MLEs of ARMA(1,1).autoregressive parameter ϕ, moving average parameter θ, and var σ^2. I generate 1000 realization for 3length$T = 50, 200, 500$,compare estimator using MSE, MAD,Coverage 95% CI.

given parameter in ques.

- $\phi = 0.9$ (AR parameter)
- $\theta = 0.5$ (MA parameter)
- $\sigma^2 = 1$ (Variance)

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(MASS)
```

variable name i used:

- `p`: AR parameter (ϕ)
- `t`: MA parameter (θ)
- `s2`: Variance (σ^2)
- `Ts`: Sample size (T) for simulations
- `pe`, `te`, `se`: Matrix for estimated parameter for ϕ, θ, and σ^2

now I define model parameter

```
p<-0.9
t<-0.5
s2<-1
```

Function to simulate & fit ARMA(1,1)

I define function to simulate ARMA(1,1) data and fit the model using MLE.If fitting fails without differencing, then i try to fit with differencing.

```
fit_arma<-function(n){
data<-arima.sim(n=n,list(ar=p,ma=t),sd=sqrt(s2))
tryCatch(Arima(data,order=c(1,0,1),method="ML"),
error=function(e) Arima(data,order=c(1,1,1),method="ML"))
}
```

Sample Size & Parameter Estimation

I also prepare matirx to store estimated params:

```
Ts<-c(50,200,500)
m<-1000
pe<-matrix(NA,m,3)
te<-matrix(NA,m,3)
se<-matrix(NA,m,3)
```

generating realizations

I generate 1000 realizations for each T & store estimated params in matirx:

```
for(i in 1:3){
for(j in 1:m){
fit<-fit_arma(Ts[i])
pe[j,i]<-coef(fit)["ar1"]
te[j,i]<-coef(fit)["ma1"]
se[j,i]<-fit$sigma2
}
}
```

error metrix

to calculate estimator, I calculate 3 metrix:MSE, MAD, Coverage of 95% CI.

MSE

average squares of errors:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(e_i - t)^2$$

- e_i: estimated param - t: true param value - n: no. of estimate

MAD

average magnitude of errors:

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|e_i - t|$$

coverage of 95% CI

I calculate how often true parameter falls within estimated CI:

$$\text{Coverage} = P(t \in [e - 1.96 \cdot se, e + 1.96 \cdot se])$$

se is std error of estimate.

Implementation of Metrics

I define functions to calculate the MSE, MAD, and coverage:

```
mse<-function(e,tr) mean((e-tr)^2)
mad_f<-function(e,tr) mean(abs(e-tr))
cov95<-function(e,tr,se_est){
l<-e-1.96*se_est
u<-e+1.96*se_est
mean(tr>=l&tr<=u)
}
```

Result

I then calculate result for each T & print the MSE, MAD, and coverage for ϕ,$heta$,σ^2:

```
for(i in 1:3){
mse_p<-mse(pe[,i],p)
mse_t<-mse(te[,i],t)
mse_s2<-mse(se[,i],s2)

mad_p<-mad_f(pe[,i],p)
mad_t<-mad_f(te[,i],t)
mad_s2<-mad_f(se[,i],s2)

cov_p<-cov95(pe[,i],p,sd(pe[,i]))
cov_t<-cov95(te[,i],t,sd(te[,i]))
cov_s2<-cov95(se[,i],s2,sd(se[,i]))

res<-matrix(c(mse_p,mse_t,mse_s2,
mad_p,mad_t,mad_s2,
cov_p,cov_t,cov_s2),
nrow=3,byrow=TRUE,
dimnames=list(c("MSE","MAD","Coverage"),
c("Phi","Theta","S2")))
cat("T =",Ts[i],"\n")
print(res)
cat("\n")
}
```

```
## T = 50
##                 Phi      Theta         S2
## MSE      0.01599998 0.02167566 0.04384145
## MAD      0.09082631 0.11309965 0.16397064
## Coverage 0.86900000 0.93800000 0.94800000
##
## T = 200
##                  Phi      Theta         S2
## MSE      0.001687663 0.00432137 0.01045008
## MAD      0.031045446 0.05090399 0.08164045
## Coverage 0.904000000 0.94500000 0.94800000
##
## T = 500
```

```
##                   Phi       Theta          S2
## MSE      0.0005665771 0.001737986 0.004043277
## MAD      0.0182505372 0.033093988 0.051170108
## Coverage 0.9360000000 0.943000000 0.954000000
```

so,metirx find accuracy& consistnecy of estimator.