Problem 1

Problem Statement:

For the MA(1) model $x_t = \theta w_{t-1} + w_t$, use the innovation algorithm to show that:

$$\theta_{n1} = \frac{\theta\sigma^2}{P_n^{n-1}}, \quad \theta_{nj} = 0, \quad j = 2, 3, \ldots, n.$$

Step-by-Step Solution Using the Innovation Algorithm

Step 1: MA(1) Model Setup

The MA(1) process is:

$$x_t = w_t + \theta w_{t-1}$$

where $w_t \sim$ i.i.d. $(0, \sigma^2)$.

The autocovariance function is:

- $\gamma_0 = \mathbb{E}[x_t^2] = \sigma^2(1 + \theta^2)$
- $\gamma_1 = \mathbb{E}[x_t x_{t-1}] = \theta\sigma^2$
- $\gamma_k = 0$ for $k \geq 2$

Step 2: Innovation Algorithm Summary

Let $\nu_n = x_n - \hat{x}_n$ be the innovation (forecast error), and $P_n = \text{Var}(\nu_n)$ its variance.

In the innovation algorithm, the best linear predictor of x_n based on past innovations is:

$$\hat{x}_n = \sum_{j=1}^{n-1} \theta_{nj} \nu_j$$

Our goal is to compute $\theta_{n1}, \theta_{n2}, \ldots, \theta_{nn-1}$.

Step 3: Structure of MA(1)

Since the MA(1) model only has correlation at lag 1 ($\gamma_1 = \theta\sigma^2$), we expect:

$$\theta_{nj} = 0 \quad \text{for } j = 2, \ldots, n-1$$

Only θ_{n1} can be nonzero.

Step 4: Applying the Innovation Formula

From the innovation algorithm:

$$\theta_{n1} = \frac{\text{Cov}(x_n, \nu_1)}{P_1}$$

Since $\nu_1 = x_1$, and:

- $x_n = w_n + \theta w_{n-1}$
- $x_1 = w_1 + \theta w_0$

Only lag-1 terms overlap:

$$\text{Cov}(x_n, x_{n-1}) = \theta\sigma^2 \Rightarrow \text{Cov}(x_n, \nu_1) = \theta\sigma^2$$

Thus:

$$\theta_{n1} = \frac{\theta\sigma^2}{P_1} = \frac{\theta\sigma^2}{P_n^{n-1}}$$

Final Answer

$$\boxed{\theta_{n1} = \frac{\theta\sigma^2}{P_n^{n-1}}, \quad \theta_{nj} = 0 \text{ for } j = 2, \ldots, n}$$

Problem 2 - 3.10(a)

```
library(astsa)
(reg <- ar.ols(cmort, order = 2, demean = FALSE, intercept = TRUE))
```

```
##
## Call:
## ar.ols(x = cmort, order.max = 2, demean = FALSE, intercept = TRUE)
##
## Coefficients:
##      1       2
## 0.4286  0.4418
##
## Intercept: 11.45 (2.394)
##
## Order selected 2  sigma^2 estimated as  32.32
```

Problem 2 - 3.10(b)

```
predict(reg, n.ahead = 4)
```

```
## $pred
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 44)
## Frequency = 52
```

```
## [1] 87.59986 86.76349 87.33714 87.21350
##
## $se
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 44)
## Frequency = 52
## [1] 5.684848 6.184973 7.134227 7.593357
```

Problem 3 - 3.21

```
phi <- rep(NA, 10)
theta <- rep(NA, 10)
sigma2 <- rep(NA, 10)
for (i in 1:10){
  x <- arima.sim(n = 200, list(ar = .9, ma = .5))
  fit <- arima(x, order = c(1, 0, 1))
  phi[i] <- fit$coef[1]
  theta[i] <- fit$coef[2]
  sigma2[i] <- fit$sigma2
}
cbind("phi" = phi, "theta" = theta, "sigma2" = sigma2)
```

```
##             phi     theta    sigma2
##  [1,] 0.9042891 0.4109222 1.0180938
##  [2,] 0.8713635 0.4649760 1.0124829
##  [3,] 0.9177908 0.3790254 1.0764262
##  [4,] 0.8537655 0.5267092 1.0426955
##  [5,] 0.9030514 0.4752496 1.0955291
##  [6,] 0.9159962 0.5368575 0.9812926
##  [7,] 0.8268371 0.5088409 1.1630592
##  [8,] 0.8253408 0.6679340 0.9547600
##  [9,] 0.9049109 0.6071885 1.0159178
## [10,] 0.8570367 0.4837226 0.9266036
```

These estimates are approximately around the same as the true values.

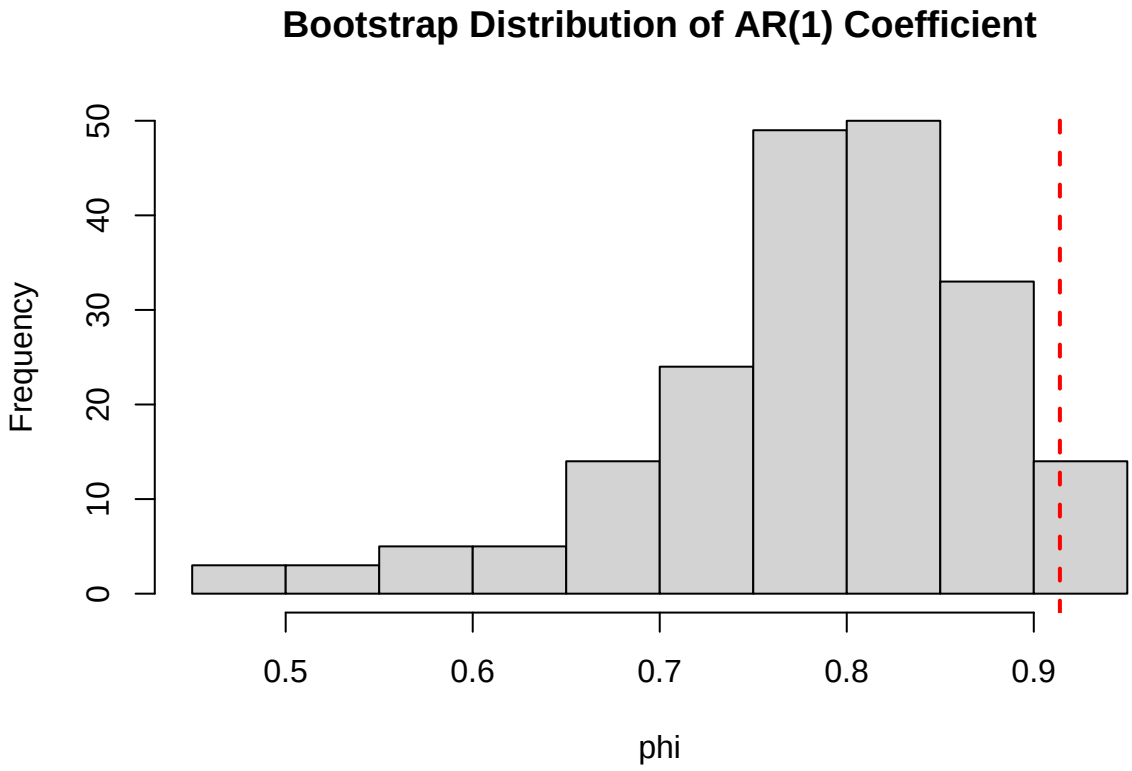
Problem 4 - 3.22

```
set.seed(123)  # for reproducibility
x <- arima.sim(list(ar = 0.99), n = 50)
fit <- ar.yw(x, order = 1)
phi.hat <- fit$ar
nboot <- 200
resids <- na.omit(fit$resid)  # remove NA
phi.star <- numeric(nboot)

for (i in 1:nboot) {
  resid.star <- sample(resids, replace = TRUE)
  x.star <- numeric(50)
  x.star[1] <- x[1]
  for (t in 1:49) {
```

```
    x.star[t+1] <- phi.hat * x.star[t] + resid.star[t]
  }

  # Estimate phi from the bootstrapped series
  fit.star <- try(ar.yw(x.star, order = 1), silent = TRUE)
  if (!inherits(fit.star, "try-error") && length(fit.star$ar) > 0) {
    phi.star[i] <- fit.star$ar
  } else {
    phi.star[i] <- NA  # mark failed bootstrap
  }
}

# Remove failed replicates
phi.star <- na.omit(phi.star)

# Compare distributions
summary(phi.star)
```

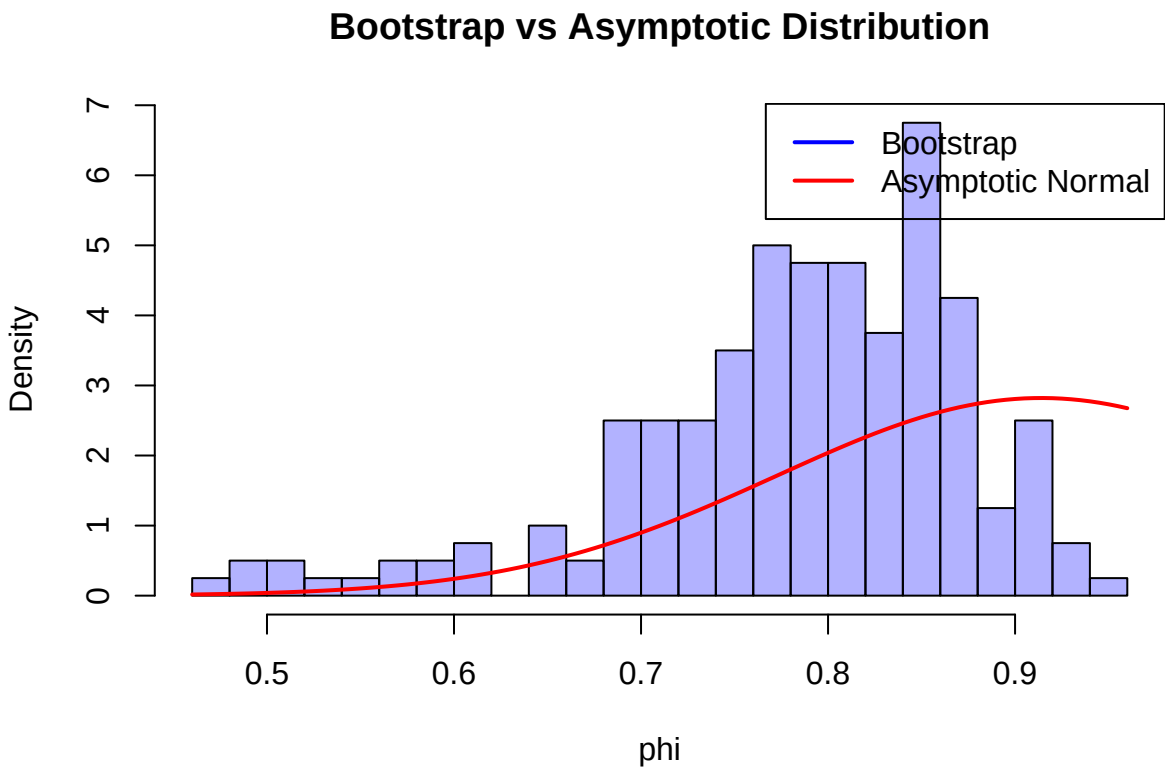
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4606  0.7405  0.7926  0.7833  0.8485  0.9446
```

```
hist(phi.star, main = "Bootstrap Distribution of AR(1) Coefficient", xlab = "phi")
abline(v = phi.hat, col = "red", lwd = 2, lty = 2)
```




```
# Approximate standard error under asymptotic theory
se.asymp <- sqrt(1 / length(x))
phi.norm <- rnorm(200, mean = phi.hat, sd = se.asymp)

# Overlay the bootstrap vs normal
hist(phi.star, probability = TRUE, col = rgb(0,0,1,0.3), main = "Bootstrap vs Asymptotic Distribution",
```

```r
curve(dnorm(x, mean = phi.hat, sd = se.asymp), col = "red", lwd = 2, add = TRUE)
legend("topright", legend = c("Bootstrap", "Asymptotic Normal"), col = c("blue", "red"), lwd = 2)
```




```r
# Summary stats from bootstrap
mean.phi <- mean(phi.star)
sd.phi <- sd(phi.star)

cat("Estimated phi from original data:", phi.hat, "\n")
```

```
## Estimated phi from original data: 0.9139702
```

```r
cat("Bootstrap mean:", mean.phi, "\n")
```

```
## Bootstrap mean: 0.7833266
```

```r
cat("Bootstrap standard deviation:", sd.phi, "\n")
```

```
## Bootstrap standard deviation: 0.09201894
```

```r
# Percentile bootstrap 95% CI
ci.percentile <- quantile(phi.star, probs = c(0.025, 0.975))
cat("95% Bootstrap CI (Percentile):\n")
```

```
## 95% Bootstrap CI (Percentile):
```

```r
print(ci.percentile)
```

```
##      2.5%     97.5%
## 0.5289996 0.9114811
```

```r
# Asymptotic distribution parameters
asymp.mean <- phi.hat
asymp.sd <- sqrt((1 - phi.hat^2) / length(x))
```

```r
cat("Asymptotic Mean of phi:", asymp.mean, "\n")
```

```
## Asymptotic Mean of phi: 0.9139702
```

```r
cat("Asymptotic SD of phi:", asymp.sd, "\n")
```

```
## Asymptotic SD of phi: 0.05738615
```

```r
# Asymptotic standard error
se.asymp <- sqrt((1 - phi.hat^2) / length(x))

# 95% asymptotic CI
ci.asymp <- phi.hat + c(-1.96, 1.96) * se.asymp
cat("95% Asymptotic CI:\n")
```

```
## 95% Asymptotic CI:
```

```r
print(ci.asymp)
```

```
## [1] 0.8014933 1.0264470
```

Comparison:

- The **bootstrap distribution** is wider and shows more variability than the **asymptotic normal** distribution.
- The **bootstrap CI** is more conservative and fully contained within the asymptotic CI.
- The **bootstrap mean** is lower than $\hat{\phi}$, indicating potential finite-sample bias.
- The **asymptotic CI** includes values above 1, which is problematic for stationarity — this highlights the limitations of normal-based inference in small samples with high persistence.

In summary, the bootstrap distribution provides a more realistic reflection of uncertainty under these conditions.