Resolución

1.

```
S <- matrix(c(
  2.45, 0.57, 2.46, -0.21,  0.76, 1.32, 1.18,  0.62,
  0.57, 6.19, 2.45,  2.95,  2.06, 3.55, 3.29,  2.28,
  2.46, 2.45, 5.44,  0.65,  1.52, 3.38, 2.63,  2.10,
 -0.21, 2.95, 0.65,  2.93,  1.02, 1.61, 1.36,  1.11,
  0.76, 2.06, 1.52,  1.02,  3.17, 0.15, 2.78, -0.08,
  1.32, 3.55, 3.38,  1.61,  0.15, 5.48, 1.41,  3.12,
  1.18, 3.29, 2.63,  1.36,  2.78, 1.41, 4.35,  0.52,
  0.62, 2.28, 2.10,  1.11, -0.08, 3.12, 0.52,  3.21
), nrow=8)
n<- 100
l <- function(x){
lambda_1 <- x[1:8]
lambda_2 <- x[9:12]
lambda_3 <- x[13:16]

lambda <- matrix(0, nrow = 8, ncol = 3)
lambda[, 1] <- lambda_1
lambda [1:4,2] <- lambda_2
lambda[5:8,3] <- lambda_3







psi <- diag(x[17:24])




sigma <- lambda%*%t(lambda) + psi
invSigma <- solve(sigma)

log_lk <- n*0.5*log(det(invSigma)) - n*0.5*sum(diag(S%*%invSigma))
return(log_lk)

}

fit <- optim(par=c(rep(1,24)), ##  24 parámetros: 8+4+4 (+8 varianzas error)
fn=l, method="L-BFGS-B", hessian=TRUE, control=list(fnscale=-1))

print(fit$par)
```

```
## [1]  0.7424757  2.0845139  1.8088547  0.9423816  0.9351014
## [6]  1.7729366  1.5243448  1.1051713  0.9291958 -1.0852020
## [11]  1.2018820 -0.9076569  1.2142247 -1.1991485  1.1188079
## [16] -0.9732348  1.0353646  0.6667106  0.7236182  1.2178655
## [21]  0.8213274  0.8985189  0.7746446  1.0413696
```

2.

```
H <- fit$hessian # segundas derivadas
Informacion <- - H # información = - (segundas derivadas)
VarCovar <- solve(Informacion) # varianza de los estimadores = 1/información
print(sqrt(diag(VarCovar)), digits=2) # mostrar la varianza de los estimadores

## [1] 0.15 0.21 0.20 0.16 0.17 0.20 0.18 0.17 0.14 0.15 0.16
## [12] 0.15 0.14 0.15 0.14 0.14 0.20 0.27 0.31 0.22 0.19 0.25
## [23] 0.20 0.19
```

3.

```
invS <- solve(S)
p <- ncol(S)
log_lk_saturado <- n*0.5*log(det(invS)) - n*0.5*p

G2 <- -2*(fit$value - log_lk_saturado)
par_saturado <- p*(p+1)/2
par_factorial <- length(fit$par)
gl <- par_saturado - par_factorial
p_valor <- pchisq(G2, gl, lower.tail = F)


cat(sprintf("G2 =%5.2f, gl =%1.0f, p =%3.2f\n", G2, gl, p_valor))

## G2 = 8.59, gl =12, p =0.74
```

4.

```
AIC_saturado <- -2*log_lk_saturado + 2*par_saturado
AIC_factorial <- -2*fit$value + 2*par_factorial
cat(sprintf(
"AIC, saturado =%5.2f, bifactor =%5.2f",
AIC_saturado, AIC_factorial))

## AIC, saturado =1448.04, bifactor =1432.63
```

5.
El valor de AIC del modelo bifactor es menor que el del modelo saturado, por lo que obtenemos un primer indicador de que nuestro modelo tiene un mejor ajuste.
Por otra parte, no tenemos suficiente evidencia estadística para rechazar la hipótesis de que el modelo bifactor reproduce la misma matriz que el saturado, por lo que concluimos que nuestro modelo ajusta a los datos, y que por lo tanto, es apropiado.