7-Confirmatory-Factor-Analysis-Assignment

Setup and Data Prep

This assignment uses a daily stock return dataset. The fields are companies, with MAN being manufacturing and Serv being service.

```
stock <- read.csv("https://bit.ly/3egKiMU")
# Multiplying by 100 to convert to % scale
mydata<-stock*100
head(mydata)
```

	Man1 <dbl>	Man2 <dbl>	Man3 <dbl>	Serv1 <dbl>	Serv2 <dbl>	Serv3 <dbl>	Serv4 <dbl>
1	-1.0376336	-0.2811634	0.1044327	0.4206954	-0.4319530	-1.1520534	-0.0218869
2	-1.5294139	-1.7052611	-0.2008442	1.3144198	0.3391672	-0.5959107	0.5752426
3	-0.5597689	-0.0867514	0.7647221	0.2495059	-0.4570792	-1.1096758	1.0483611
4	-0.6594511	0.8015072	0.2140318	-0.6727501	-0.3915131	0.4250358	0.5515717
5	0.8006007	-0.4137202	-0.2886513	1.1790268	1.1174818	1.3899308	-0.6460089
6	0.9088881	-0.2099296	0.6839741	0.8345575	-0.4702975	-0.1639780	-0.3244047

6 rows

Load the Sem packages:

```
install.packages("sem", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T//RtmpdN4TYA/downloaded_packages
```

```
install.packages("semPlot", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/1n/nvr79nb55tz9j4lsbrw6hsf80000gn/T//RtmpdN4TYA/downloaded_packages
```

```
library(sem)
```

```
## Warning: package 'sem' was built under R version 4.0.2
```

a) Perform confirmatory factor analysis base on two factors: manufacturing and service.

To create this model, we specify the factors we are testing for each manifest variable. We also pass in the correlation between the factors to be read out as rho, and the individual variances of each column.

```
stock_model <- specifyModel(text = "
                 Manu -> Man1, lambda1, NA
                 Manu -> Man2, lambda2, NA
                 Manu -> Man3, lambda3, NA
                 Serv -> Serv1, lambda4, NA
                 Serv -> Serv2, lambda5, NA
                 Serv -> Serv3, lambda6, NA
                 Serv -> Serv4, lambda7, NA
                 Manu <-> Serv, rho, NA
                 Man1 <-> Man1, theta1, NA
                 Man2 <-> Man2, theta2, NA
                 Man3 <-> Man3, theta3, NA
                 Serv1 <-> Serv1, theta4, NA
                 Serv2 <-> Serv2, theta5, NA
                 Serv3 <-> Serv3, theta6, NA
                 Serv4 <-> Serv4, theta7, NA
                 Manu <-> Manu, NA, 1
                 Serv <-> Serv, NA, 1
                ")
```

```
## NOTE: it is generally simpler to use specifyEquations() or cfa()
##       see ?specifyEquations
```

```
stock_sem <- sem(stock_model, cor(mydata), nrow(mydata))
summary(stock_sem)
```

```
##
##  Model Chisquare =  8.696696   Df =  13 Pr(>Chisq) = 0.7954455
##  AIC =  38.6967
##  BIC =  -81.3743
##
##  Normalized Residuals
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.5848593 -0.3589473 -0.0000011  0.0737239  0.2515302  1.1401712
##
##  R-square for Endogenous Variables
##   Man1   Man2   Man3  Serv1  Serv2  Serv3  Serv4
## 0.2400 0.4940 0.4667 0.1061 0.2889 0.2188 0.4018
##
##  Parameter Estimates
##         Estimate  Std Error  z value   Pr(>|z|)
## lambda1 0.4898527 0.03587045 13.656164 1.855024e-42 Man1 <--- Manu
## lambda2 0.7028188 0.03768934 18.647681 1.318745e-77 Man2 <--- Manu
## lambda3 0.6831474 0.03743433 18.249223 2.098718e-74 Man3 <--- Manu
## lambda4 0.3257044 0.03910335  8.329323 8.130615e-17 Serv1 <--- Serv
## lambda5 0.5374555 0.03948475 13.611725 3.410985e-42 Serv2 <--- Serv
## lambda6 0.4677928 0.03901408 11.990357 3.991827e-33 Serv3 <--- Serv
## lambda7 0.6338432 0.04079451 15.537464 1.934900e-54 Serv4 <--- Serv
## rho     0.4828152 0.04321207 11.173155 5.518540e-29 Serv <--> Manu
## theta1  0.7600445 0.03923737 19.370423 1.371273e-83 Man1 <--> Man1
## theta2  0.5060457 0.04301257 11.765065 5.908071e-32 Man2 <--> Man2
## theta3  0.5333097 0.04207769 12.674405 8.197880e-37 Man3 <--> Man3
## theta4  0.8939171 0.04268170 20.943801 2.137096e-97 Serv1 <--> Serv1
## theta5  0.7111416 0.04270614 16.651975 2.927759e-62 Serv2 <--> Serv2
## theta6  0.7811701 0.04200696 18.596205 3.448744e-77 Serv3 <--> Serv3
## theta7  0.5982426 0.04601226 13.001810 1.194812e-38 Serv4 <--> Serv4
##
##  Iterations =  17
```

b) Report the path diagram that shows coefficient estimates. Code: library(semPlot); semPaths(fitted.sem.object, “est”)

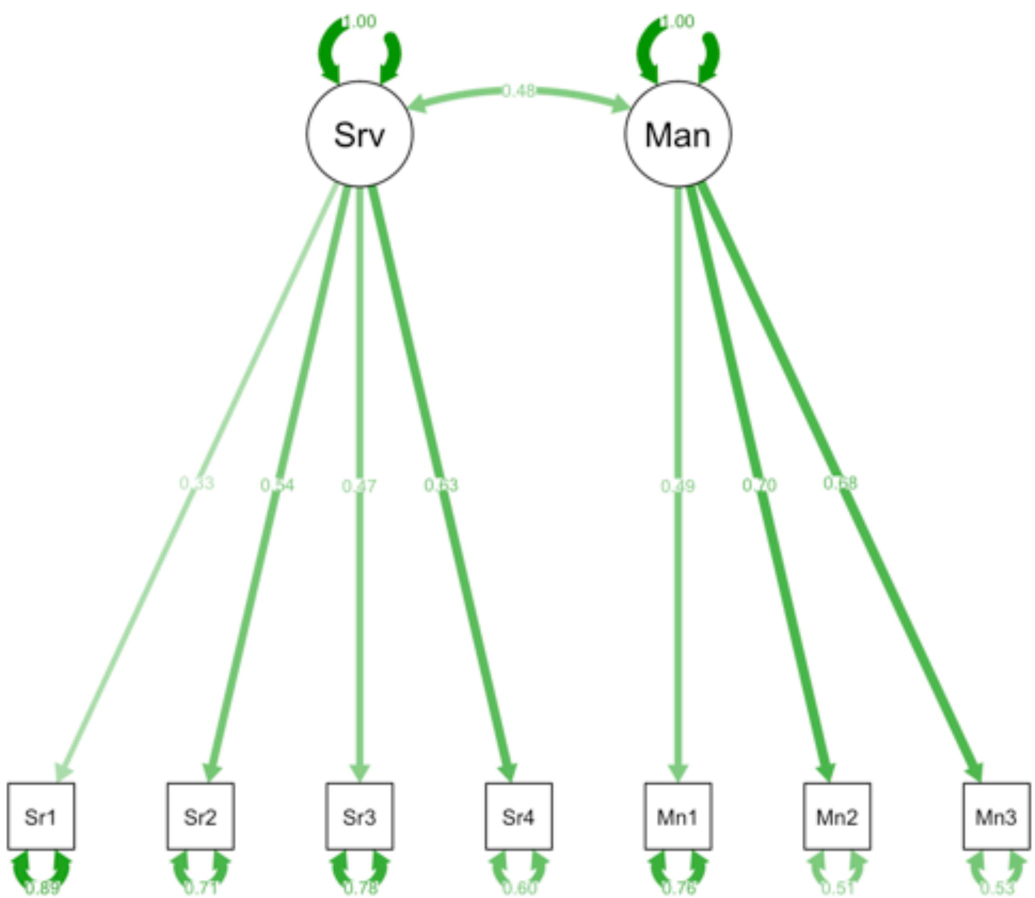
The path diagram helps us visualize the model we just created. To build it, we use the semPlot package and call the `semPaths()` function:

```
library(semPlot)
```

```
## Warning: package 'semPlot' was built under R version 4.0.2
```

```
## Registered S3 methods overwritten by 'huge':
##   method    from
##   plot.sim  BDgraph
##   print.sim BDgraph
```

```
semPaths(stock_sem, "est")
```



c) Report SRMR, GFI, and AGFI. What do you conclude? Is the model you made in part (a) approved.

The root-mean-square, goodness-of-fit, and adjusted goodness-of-fit can be pulled from the summary of our CFA analysis. The SRMR (RMSE) is < 0.05 and the goodness-of-fit tests are both > 0.95 which all imply that the data supports the CFA model.

```
options(fit.indices = c("GFI", "AGFI", "SRMR")) # Some fit indices
criteria = summary(stock_sem)
criteria$SRMR
```

```
## [1] 0.01468043
```

```
criteria$GFI
```

```
## [1] 0.9975558
```

```
criteria$AGFI
```

```
## [1] 0.9947356
```

d) Find the 95% confidence interval for the correlation between the two factors: manufacturing and service returns.

The Confidence Interval is found by adding or subtracting the product of 1.96 and the standard error from the correlation estimate to get the upper and lower bounds.

```
parameters = summary(stock_sem)

# Calculate the lower bound
conf.L = parameters$coeff[8,]$Estimate - 1.96 * parameters$coeff[8,]$'Std Error'
conf.L
```

```
## [1] 0.3981195
```

```
# Calculate the upper bound
conf.U = parameters$coeff[8,]$Estimate + 1.96 * parameters$coeff[8,]$'Std Error'
conf.U
```

```
## [1] 0.5675109
```