

Corrección y consistencia interna de cuestionarios

VIII Jornadas de Usuarios de R

Elvira Ferre Jaén (elvira@um.es)

Sección de Apoyo Estadístico, Servicio de Apoyo a la Investigación, Universidad de Murcia

17 de noviembre de 2016, Albacete



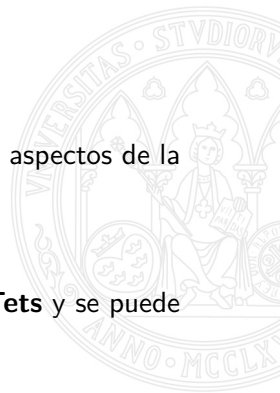
Medidas de correlación y consistencia interna



Conjunto de datos

Utilizaremos los datos de un test en el que se miden 5 aspectos de la personalidad mediante una escala Likert(5).

El conjunto de datos está disponible en **Personality Tests** y se puede descargar **aquí**



Conjunto de datos

Nos quedaremos con los 10 primeros ítems que constituyen el constructo *extroversión*

```
head( df )
```

##	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
## 1	4	4	5	4	5	5	4	3	5	5
## 2	2	4	3	3	3	3	1	1	1	1
## 3	5	5	1	2	5	5	1	1	5	5
## 4	2	1	2	2	3	2	3	2	4	1
## 5	3	5	3	3	3	5	3	5	3	1
## 6	1	1	2	2	1	3	2	2	1	1

La escala varía de la siguiente forma:

- 1 = En desacuerdo
- 3 = Neutro
- 5 = De acuerdo

Conjunto de datos

Hay que tener en cuenta la **dirección de los ítems**, todos ítems del constructo han de estar enunciados de forma afirmativa o negativa, pero no se pueden mezclar.

Imaginemos los ítems:

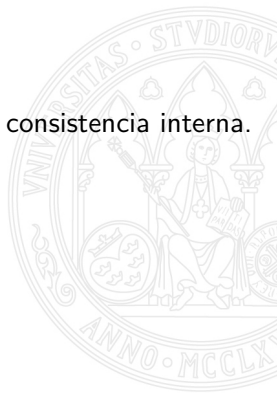
- E1: Soy el alma de la fiesta.
- E2: No me gusta hablar mucho.
- E3: Me siento cómodo/a rodeado/a de gente.

```
# Cambiamos el sentido al ítem E2  
df[, "E2"] <- 6 - df[, "E2"]
```

Medidas de consistencia interna

Existe una amplia variedad de medidas para evaluar la consistencia interna. Presentamos cinco de ellas:

- Average inter-item correlation
- Average item-total correlation
- Split-half reliability
- Cronbach's alpha
- Composite reliability



Average Inter-item Correlation



Average Inter-item Correlation

Promedio de correlación entre ítems. Pasos:

- 1 Estimar la correlación entre cada par de elementos
- 2 Hallar la correlación media de cada ítem con el resto
- 3 Calcular la media de todas estas correlaciones.



Average Inter-item Correlation en R (1)

Utilizamos el paquete `corr`.

```
library( corr )
correlate( df )
```

rowname	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
E1		.45	.50	.52	.54	.37	.64	.45	.53	.49
E2	.45		.48	.55	.59	.57	.47	.38	.40	.45
E3	.50	.48		.49	.62	.33	.57	.42	.48	.50
E4	.52	.55	.49		.51	.47	.50	.45	.46	.52
E5	.54	.59	.62	.51		.50	.62	.39	.49	.55
E6	.37	.57	.33	.47	.50		.37	.33	.33	.41
E7	.64	.47	.57	.50	.62	.37		.40	.53	.52
E8	.45	.38	.42	.45	.39	.33	.40		.60	.43
E9	.53	.40	.48	.46	.49	.33	.53	.60		.46
E10	.49	.45	.50	.52	.55	.41	.52	.43	.46	

Average Inter-item Correlation en R (2)

```
co <- correlate( df )
inter_item <- colMeans( co[, 2:11], na.rm = TRUE )
inter_item
```

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
0.4984	0.4828	0.4866	0.4992	0.5335	0.409	0.5133	0.427	0.4732	0.4814

Podemos ver que E5 y E7 son los que más fuerte correlacionan.

Average Inter-item Correlation en R (3)

Para obtener la correlación media entre los ítems basta con calcular la media de estos valores:

```
mean( inter_item )
```

```
[1] 0.4804446
```

Están comúnmente aceptados valores entre 0.15 y 0.5.

Average item-total correlation



Average item-total correlation

La medida de correlación media ítem-total surge ante el problema de construir una **única cantidad útil para cada individuo** que permita comparar ese individuo con el resto de la población.

- 1 Calcular la puntuación total para cada individuo (media de sus respuestas).
- 2 Hallar la correlación de los ítems pero centrándonos en las puntuaciones totales.
- 3 Calcular la media de todas las correlaciones de las puntuaciones totales.

Average item-total correlation en R (1)

```
df$score <- rowMeans( df )
head( df )
```

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	score
4	4	5	4	5	5	4	3	5	5	4.4
2	4	3	3	3	3	1	1	1	1	2.2
5	5	1	2	5	5	1	1	5	5	3.5
2	1	2	2	3	2	3	2	4	1	2.2
3	5	3	3	3	5	3	5	3	1	3.4
1	1	2	2	1	3	2	2	1	1	1.6

Average item-total correlation en R (2)

```
df %>% correlate() %>% focus( score )
```

rowname	score
E1	0.7520849
E2	0.7297506
E3	0.7350644
E4	0.7485092
E5	0.7945943
E6	0.6367799
E7	0.7768180
E8	0.6640914
E9	0.7273987
E10	0.7306038



Average item-total correlation en R (3)

Para obtener la correlación media entre el total de los ítems basta con calcular la media de los valores de la tabla anterior:

```
mean( item_total$score )
```

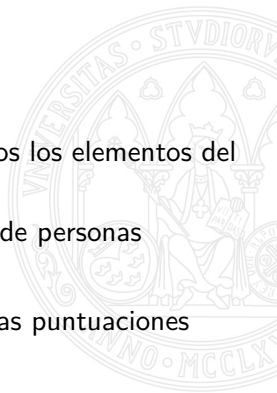
```
[1] 0.7295695
```


Split-Half Reliability



Split-Half Reliability

- **Dividimos** aleatoriamente en **dos conjuntos** todos los elementos del cuestionario que miden el mismo constructo
- **Aplicamos todo el instrumento** a una muestra de personas
- Calculamos la puntuación total de cada conjunto
- La fiabilidad es simplemente la correlación entre las puntuaciones totales de cada mitad



Split-Half Reliability en R

```
# Calculamos los conjuntos aleatorios
sel <- sample(1:10, 5)
score_1 <- rowMeans( df[, sel] ) # conjunto 1
score_2 <- rowMeans( df[, -sel] ) # conjunto 2

# correlación entre los conjuntos
r <- cor( score_1, score_2 ); r
```

```
[1] 0.7970573
```

Cronbach's alpha



Cronbach's alpha

- Imaginemos que calculamos una vez la *fiabilidad de dos mitades* y luego dividimos aleatoriamente los ítems en otro conjunto de mitades divididas y recalculamos la fiabilidad.
- Seguimos haciendo esto hasta que hayamos calculado todos los posibles estimaciones de fiabilidad por este método
- Si calculamos la media de todas estas medidas, su valor es matemáticamente equivalente al **coeficiente Cronbach**.

Cronbach's alpha en R (1)

Utilizaremos la función `alpha()` el paquete `psych`.

```
psych::alpha( df )$total$std.alpha
```

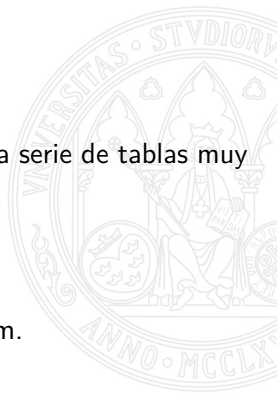
```
[1] 0.9242082
```



Cronbach's alpha en R (2)

Además del valor del Alfa de Cronbach nos devuelve una serie de tablas muy interesantes:

- Una tabla con los principales estadísticos
- Una tabla de frecuencias y valores perdidos
- Una tabla de fiabilidad cuando eliminamos un ítem.



Cronbach's alpha en R (3)

```
psych::alpha( df )$alpha.drop
```

Tabla 5: Fiabilidad al eliminar un ítem

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	alpha se
E1	0.9123	0.9174	0.9648	0.5262	11.11	0.005882
E2	0.9137	0.9185	0.9641	0.5298	11.27	0.005764
E3	0.9132	0.9182	0.9647	0.5289	11.23	0.005823
E4	0.9124	0.9174	0.9654	0.5261	11.1	0.005862
E5	0.9098	0.9149	0.9673	0.5182	10.76	0.006042
E6	0.9186	0.9234	0.9621	0.5466	12.06	0.005453
E7	0.9114	0.9163	0.9612	0.5226	10.95	0.005956
E8	0.9174	0.9222	0.9621	0.5424	11.85	0.005533
E9	0.9145	0.9191	0.9601	0.5318	11.36	0.005717
E10	0.9137	0.9186	0.9617	0.53	11.28	0.005788
score	0.9022	0.9024	0.9038	0.4804	9.247	0.006511

Composite reliability



Composite reliability (1)

- Esta medida se basa en las cargas de los factores de un análisis factorial confirmatorio (AFC).
- Definimos el factor **extraversión** y usaremos las cargas del AFC para estimar la consistencia interna.

```
library( lavaan )
modelo <- " extraversion =~ E1 + E2 + E3 + E4 + E5 +
           E6 + E7 + E8 + E9 + E10"
fit <- cfa( modelo, data = df )
```

Composite reliability (2)

Utilizaremos las cargas factoriales estandarizadas:

```
sl <- standardizedSolution( fit )  
sl <- sl$est.std[ sl$op == "=~"]
```

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
0.7267	0.6903	0.7155	0.7119	0.7841	0.5792	0.7589	0.5963	0.6726	0.694

Composite reliability (3)

```
# Calculamos la varianza residual para cada ítem
```

```
re <- 1 - sl^2
```

```
# Calculamos la fiabilidad compuesta
```

```
sum( sl )^2 / ( sum( sl )^2 + sum( re ) )
```

```
[1] 0.9029523
```

Gráficos de correlación



Matriz de correlación

Estamos acostumbrados a estudiar las correlaciones con matrices:

rowname	Ozone	Solar.R	Wind	Temp	Month	Day
Ozone		.35	-.60	.70	.16	-.01
Solar.R	.35		-.06	.28	-.08	-.15
Wind	-.60	-.06		-.46	-.18	.03
Temp	.70	.28	-.46		.42	-.13
Month	.16	-.08	-.18	.42		-.01
Day	-.01	-.15	.03	-.13	-.01	

Una forma más ilustrativa de hacerlo es **gráficamente**

Función `network_plot()`

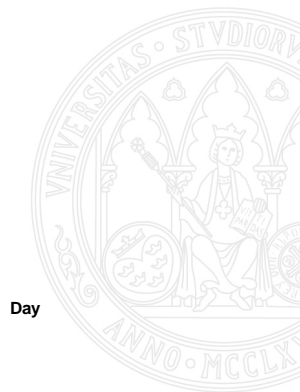
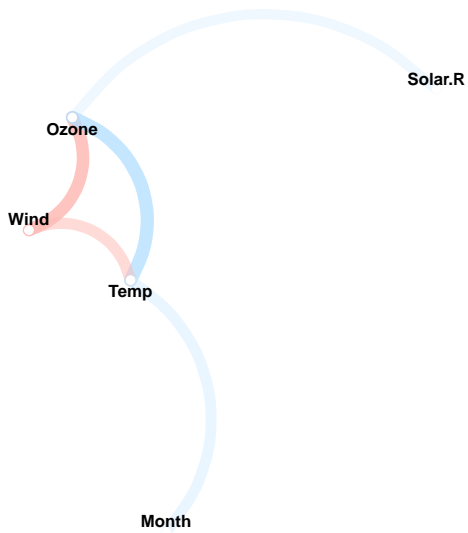


network_plot()

Utilizaremos la función network_plot() del paquete corrr

```
library( corrr )  
cor <- correlate( airquality )  
network_plot( cor )
```


network_plot()



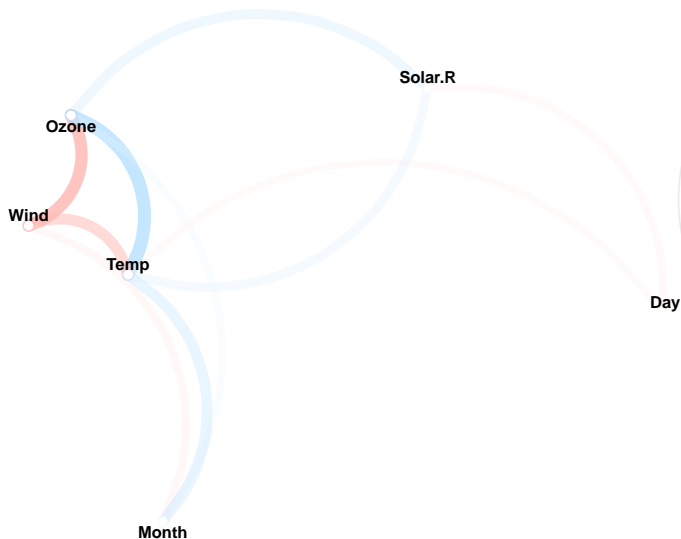
Day

Argumento `min_cor`

- Podemos observar que no aparecen todas las correlaciones en el gráfico.
- Esto se debe al argumento `min_cor` que por defecto toma valor 0.3.

```
network_plot( cor, min_cor = 0.1 )
```

Argumento min_cor



Función corrplot()



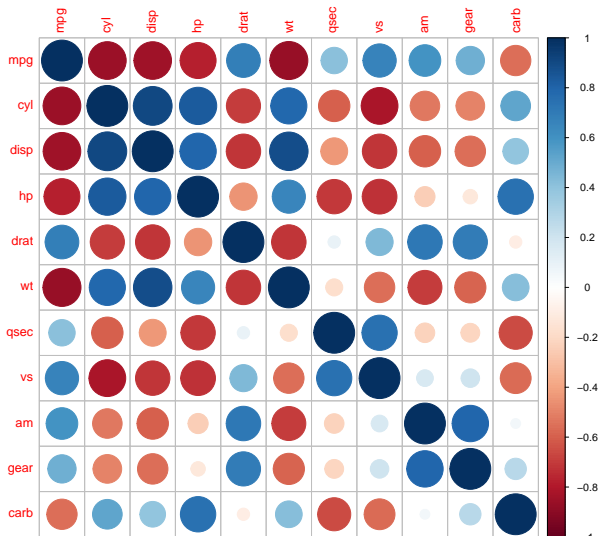
corrplot()

Utilizaremos la función `corrplot()` del paquete `corrplot`

```
library( corrplot )
```

```
co <- cor( mtcars )  
corrplot( co )
```

corrrplot()

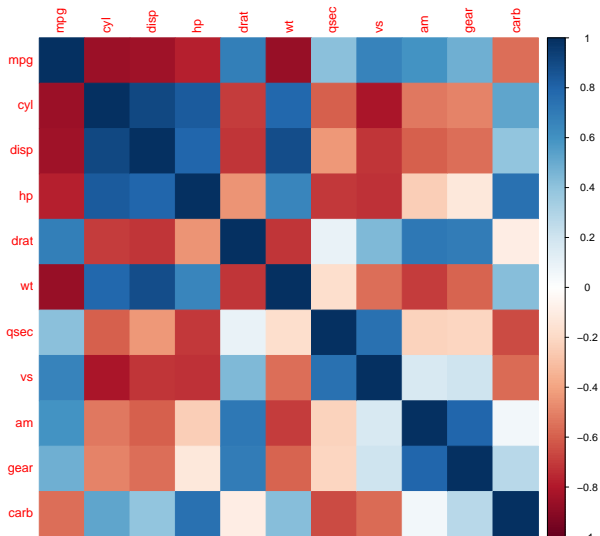


`corrplot()` de color

Podemos representar las correlaciones mediante colores según su valor

```
corrplot( co, method="color")
```

corrrplot() de color



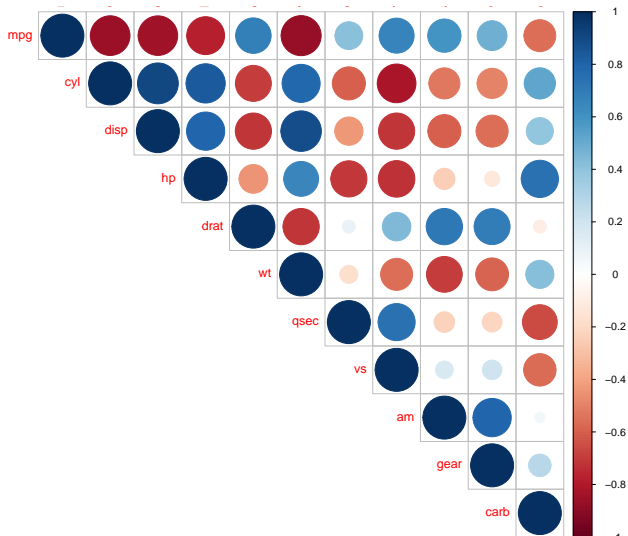
`corrplot()` superior

Otra opción es quedarnos sólo con la parte superior de la matriz

```
corrplot( co, type = "upper" )
```



corrrplot() superior



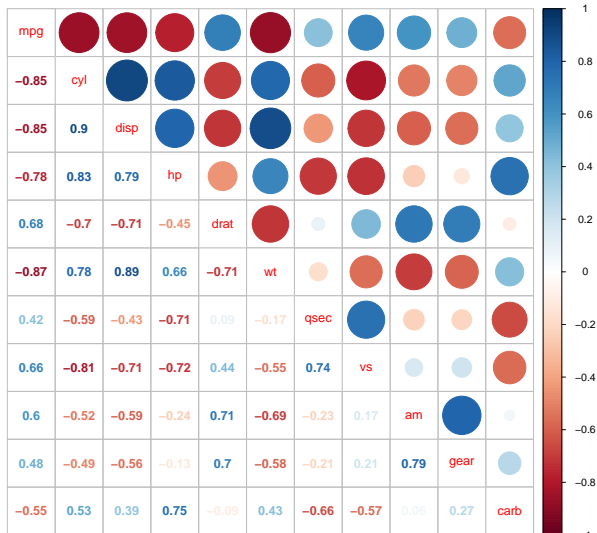
`corrplot()` mixto

O mezclar valores numéricos y puntos

```
corrplot.mixed( co )
```



corrplot() mixto



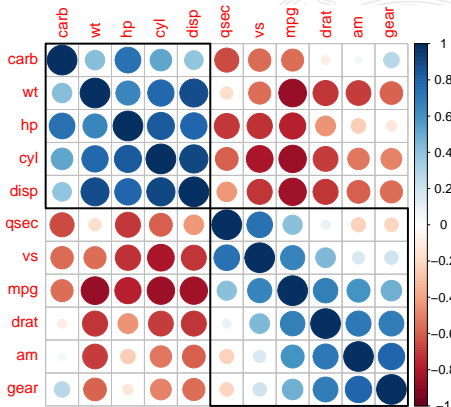
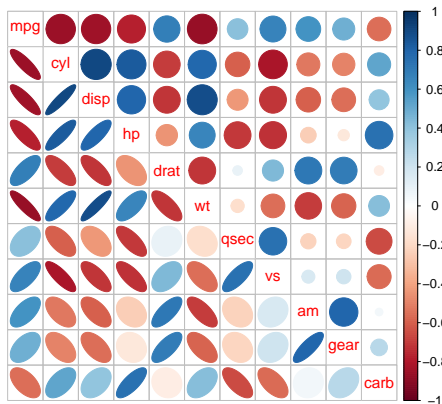
Más opciones de corrplot()

Podéis encontrar muchas más opciones en la **viñeta del paquete**

Algunas de ellas son:

```
corrplot.mixed( co, lower="ellipse", upper="circle")  
corrplot( co, order="hclust", addrect=2)
```

Más opciones de corrrplot()



Muchas gracias

Corrección y consistencia interna de cuestionarios

VIII Jornadas de Usuarios de R

Elvira Ferre Jaén (elvira@um.es)

Sección de Apoyo Estadístico, Servicio de Apoyo a la Investigación, Universidad de Murcia

17 de noviembre de 2016, Albacete

