{desafío}
latam_

Dimensionalidad y agrupación_

Sesión Presencial 1

{desafío}
latam_

Alcances de la lectura asignada

Alcances de la unidad

- *Entender el problema de la "maldición de la dimensionalidad" y sus implicancias para el modelo.*
- *Conocer la aproximación psicométrica del Principal Component Analysis y el Análisis Factorial.*
- *Implementar algoritmos de reducción de dimensiones (Principal Components Analysis) y de reconocimiento de estructuras latentes (Análisis Factorial) con* `factor_analysis`, `sklearn`.
- *Utilizar técnicas para identificar patrones de datos perdidos.*
- *Implementar algoritmos de agrupación (k-Means).*

Activación de Conceptos

- En la unidad anterior aprendimos sobre métricas de clasificación y métodos de validación cruzada.
- ¡Pongamos a prueba nuestros conocimientos!
- Las preguntas van en subslides.

¿Por qué las métricas de reducción de errores como el MSE no nos sirven para la clasificación?

1. Porque asumen que la función candidata es lineal en el plano cartesiano.

- Porque las clases estimadas son discretas y no se puede estimar errores respecto a una recta.
- Porque en el problema de clasificación buscamos minimizar los falsos positivos y negativos, cosa ignorada por los modelos lineales.

¿Cuál de las siguientes frases resume de mejor manera la validación cruzada?

1. La validación cruzada permite evaluar la función candidata en datos de la muestra.

- La validación cruzada simula el comportamiento predictivo de la función candidata en nuevos datos.
- La validación cruzada facilita la extrapolación de la función candidata

¿Cuál es el principal rasgo del estadístico F1?

1. Permite generar una aproximación a la tasa de verdaderos positivos sobre la tasa de clasificaciones totales.

- Considera tanto Precision como Recall y penaliza cuando uno de los dos valores es 0.
- Permite generar una aproximación a la tasa de verdaderos positivos sobre la tasa de clasificaciones correctas.

¿En qué se diferencia KFold CV con Leave-One-Out?

1. En la cantidad de bases creadas.

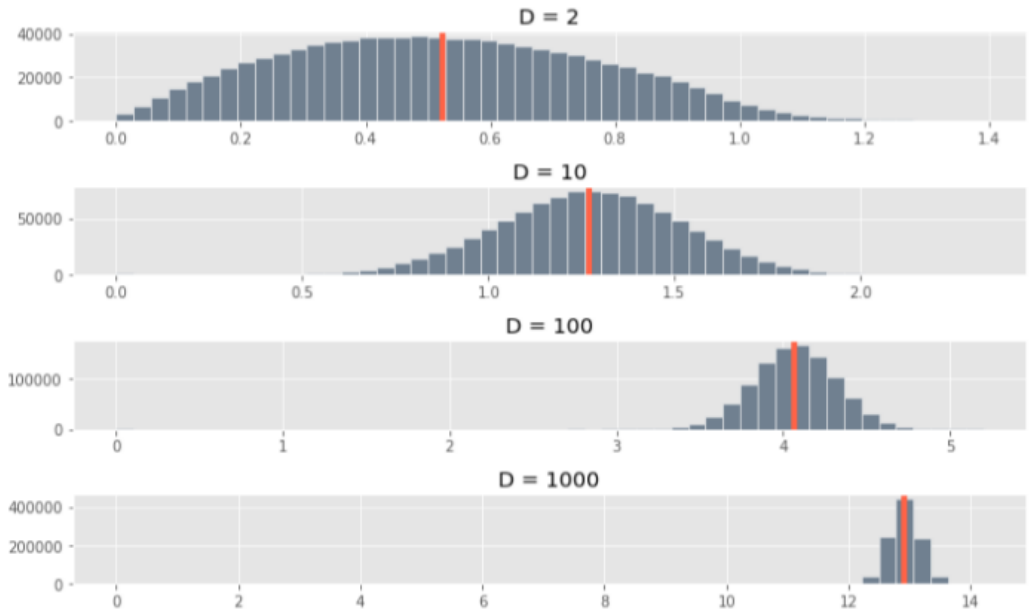
- En el método de iteración.
- En las métricas de desempeño utilizadas.

Dimensionalidad

{desafío}
latam_

La distancia promedio en dimensiones altas

In [4]:
```
plt.figure(figsize=(10,6))
for i, e in enumerate([2, 10, 100, 1000]):
    plt.subplot(4, 1, i+1)
    gfx.sim_distance(e)
    plt.tight_layout()
```



{desafío}
latam_

Análisis Factorial

{desafío}
latam_

¿Qué es?

- **Objetivo**: Extraer variabilidad conjunta y convertirlas en factores **latentes**.
- Se representa mediante un sistema de ecuaciones:

$$
\begin{aligned}
X_1 &= \lambda_{10} + \lambda_{11} f_1 + \ldots \lambda_{1k} f_k + \varepsilon_1 \\
X_2 &= \lambda_{20} + \lambda_{21} f_1 + \ldots \lambda_{2k} f_k + \varepsilon_2 \\
&\vdots \\
X_N &= \lambda_{n0} + \lambda_{n1} f_1 + \ldots \lambda_{nk} f_k + \varepsilon_n
\end{aligned}
$$

Diferencias entre Análisis Factorial y PCA

- **PCA** ⇝ orientado a maximizar la varianza de los eigenvalues.
- **Análisis Factorial** ⇝ orientado a generar componentes conceptualmente significativos.

Implementando un análisis factorial con `factor_analyzer`

- El análisis factorial usualmente se realiza en **baterías de datos prediseñadas**.
- En caso que nuestros items no tengan la misma escala, podemos estandarizarlas:

```
df.loc[:, 'conjunto':'variables'].applymap(lambda x: stats.zs
core(x))
```

- Antes de generar nuestro análisis, debemos asegurar que nuestra batería de preguntas cumpla con los requisitos:
    - **Existencia de covarianza entre los elementos**.

Esferacidad de Barlett

- Pregunta a responder ⇝ ¿Es nuestra matriz de correlación una matriz de identidad?
  - Matriz de identidad:

$$\mathbb{I}_n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In [6]:
```
import factor_analyzer as fact
# Con nuestra batería separada
fact.calculate_bartlett_sphericity(trust_df)
```

Out[6]:
```
(6630.891612219014, 0.0)
```

- Si el `p-value` asociado es menor a 0.05, concluímos que nuestra matriz no es de identidad y hay covarianza entre los items.

Kaiser-Meyer-Olkin

- Pregunta a responder ⇝ ¿Son las correlaciones parciales de la matriz cercanas a 0?

In [7]:
```
# extraemos el kmo general
fact.calculate_kmo(trust_df)[1]
```

Out[7]:
```
0.9275552166414559
```

- Valores sobre .7 = Estamos habilitados para seguir con el análisis.

In [8]:
```
fact.calculate_kmo(trust_df)[0][:7].T
```

Out[8]:

	I.Catolica	I.Evangelica	FFAA	Justicia	Prensa	Television	Sindi
KMO	0.938611	0.927266	0.931852	0.952914	0.919814	0.885661	0.93

In [9]:
```
fact.calculate_kmo(trust_df)[0][7:].T
```

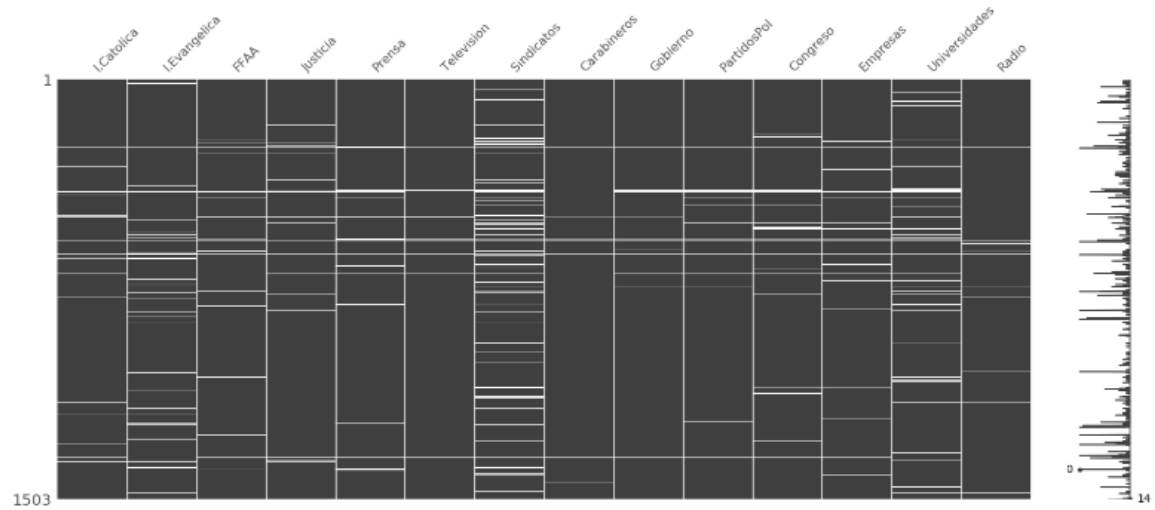
Out[9]:

	Carabineros	Gobierno	PartidosPol	Congreso	Empresas	Universidades
KMO	0.911312	0.939891	0.913019	0.919379	0.942421	0.951845

Visualizando la matriz de perdidos

In [10]:
```
import missingno as msngo
msngo.matrix(trust_df.replace([8, 9], [np.nan, np.nan]))
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1169379e8>
```



{desafío}
latam_

Iniciando nuestro análisis

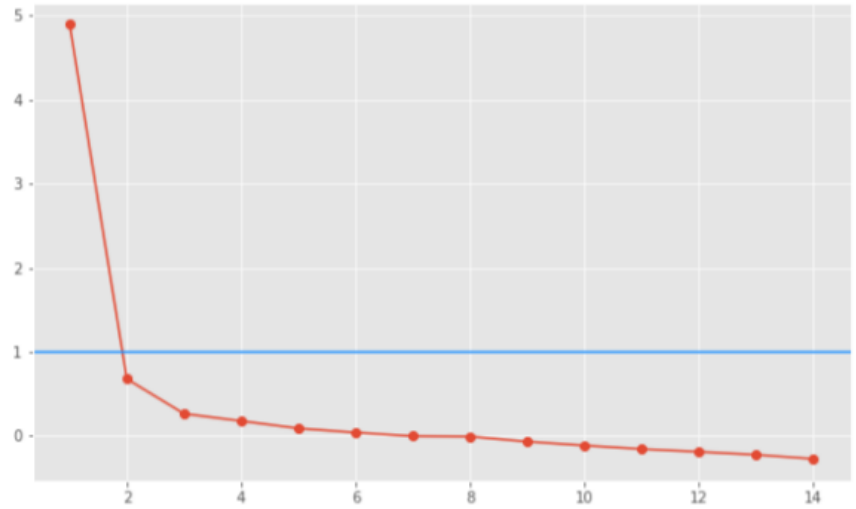
In [11]:
```
factor_trust_results = fact.FactorAnalyzer()
factor_trust_results.analyze(data=trust_df, n_factors=2, method='varimax')
```

Scree plot

- ¿Cuántas dimensiones debemos retener en función de los eigenvalues?

In [13]:
```
plt.figure(figsize=(10,6))
fact_scree = factor_trust_results.get_eigenvalues()[0]
plt.plot(fact_scree.index+1, fact_scree.values,'o-')
plt.axhline(1, color='dodgerblue')
```

Out[13]:
```
<matplotlib.lines.Line2D at 0x1149c5940>
```

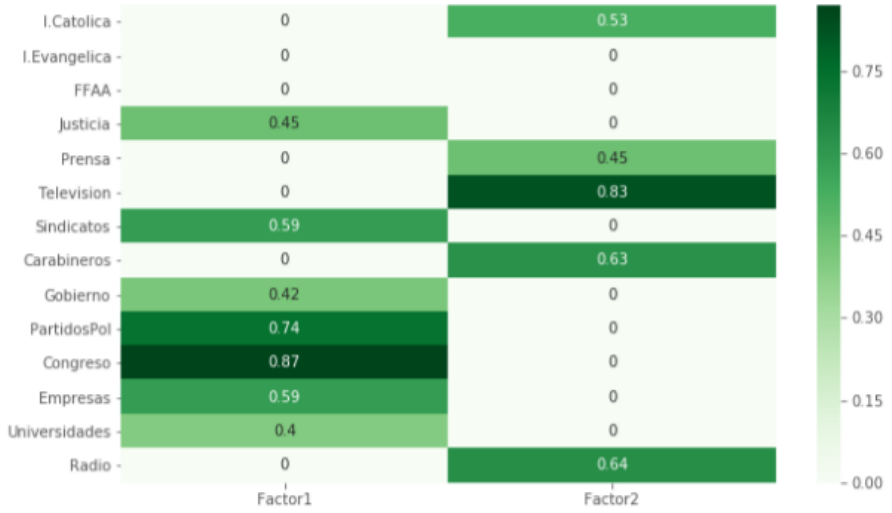


{desafío}
latam_

Extracción de factores

In [14]:
```
plt.figure(figsize=(10,6))
factor_loadings = factor_trust_results.loadings
factor_loadings = factor_loadings.applymap(lambda x: np.where(x < .4, 0, round(x, 2)))
sns.heatmap(factor_loadings, cmap='Greens', annot=True)
```

Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1173b0278>
```



{desafío}
latam_