Week 8: Unsupervised Scale Measurement I: Interval-Level Indicators

POLS0013 Measurement in Data Science

Dr. Julia de Romémont

Academic Year 24-25

UCL Departement of Political Science

Last five lectures

Supervised Measurement

- Measurement tasks where...
  1. we had a collection of indicators and
  2. wanted to figure out *how to put them together* **to measure a target concept** (on a scale or as categories).
- We *supervised* the *how*, by using some data on how the indicators relate to the target concept to determine how to put them together into an index.
- In other words, we use pre-existing knowledge or pre-existing data to determine the relative *weights* of the indicators
- We covered various ways in which we can *purposely* connect indicators and a target concept to create a measure.

This and the next two lectures are about...

Unsupervised Measurement

- Measurement tasks where...
  1. we have a collection of indicators and
  2. want to use *how they relate to each other* **as a measure of a target concept** (on a scale or as categories).
- The *how* is now *unsupervised*, in the sense that we will rely on **covariation** between indicators to determine how they relate to a target concept.
- The question is no longer "what is the best way to measure this thing from these data?" but instead "*what is the thing we can measure best from these data*?

This lecture

From supervision to covariation

Principal Component Analysis (PCA)

Exploratory Factor Analysis (EFA)

From covariation to interpretation

From supervision to covariation

Attitudes towards US government spending

- Imagine that we want to measure something about the **attitudes towards government spending** of respondents in the US
- The American National Election Study (ANES) runs surveys of voters before and after each US presidential election since 1948
- Includes a battery of 8 questions about respondents' attitudes towards current levels of government spending
- We will use data from the 2020 post-election survey

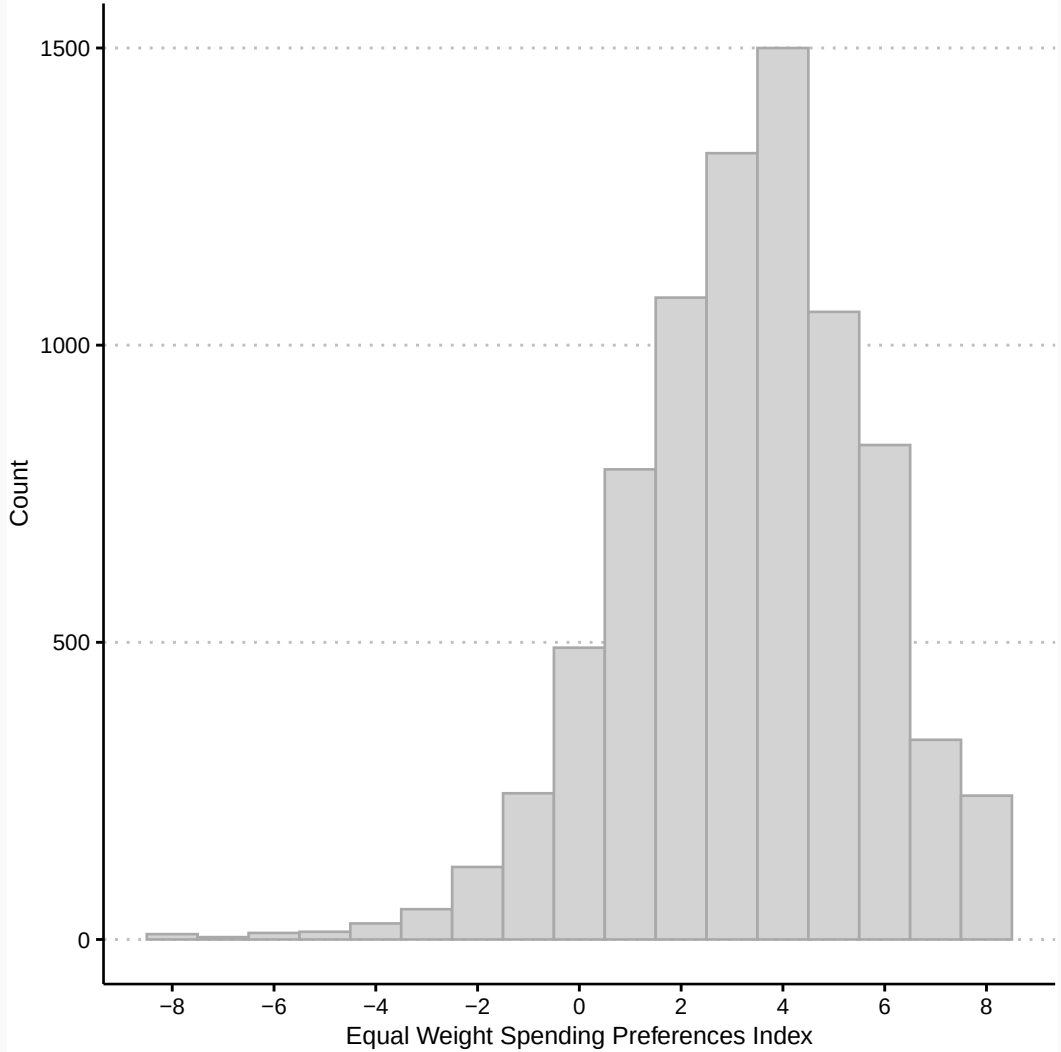
8 questions about government spending

1. Should federal spending on Social Security be increased, decreased, or kept the same?
2. Should federal spending on public schools be increased, decreased, or kept the same?
3. Should federal spending on tightening border security to prevent illegal immigration be increased, decreased, or kept the same?
4. Should federal spending on dealing with crime be increased, decreased, or kept the same?
5. Should federal spending on welfare programs be increased, decreased, or kept the same?
6. Should federal spending on building and repairing highways be increased, decreased, or kept the same?
7. Should federal spending on aid to the poor be increased, decreased, or kept the same?
8. Should federal spending on protecting the environment be increased, decreased, or kept the same?

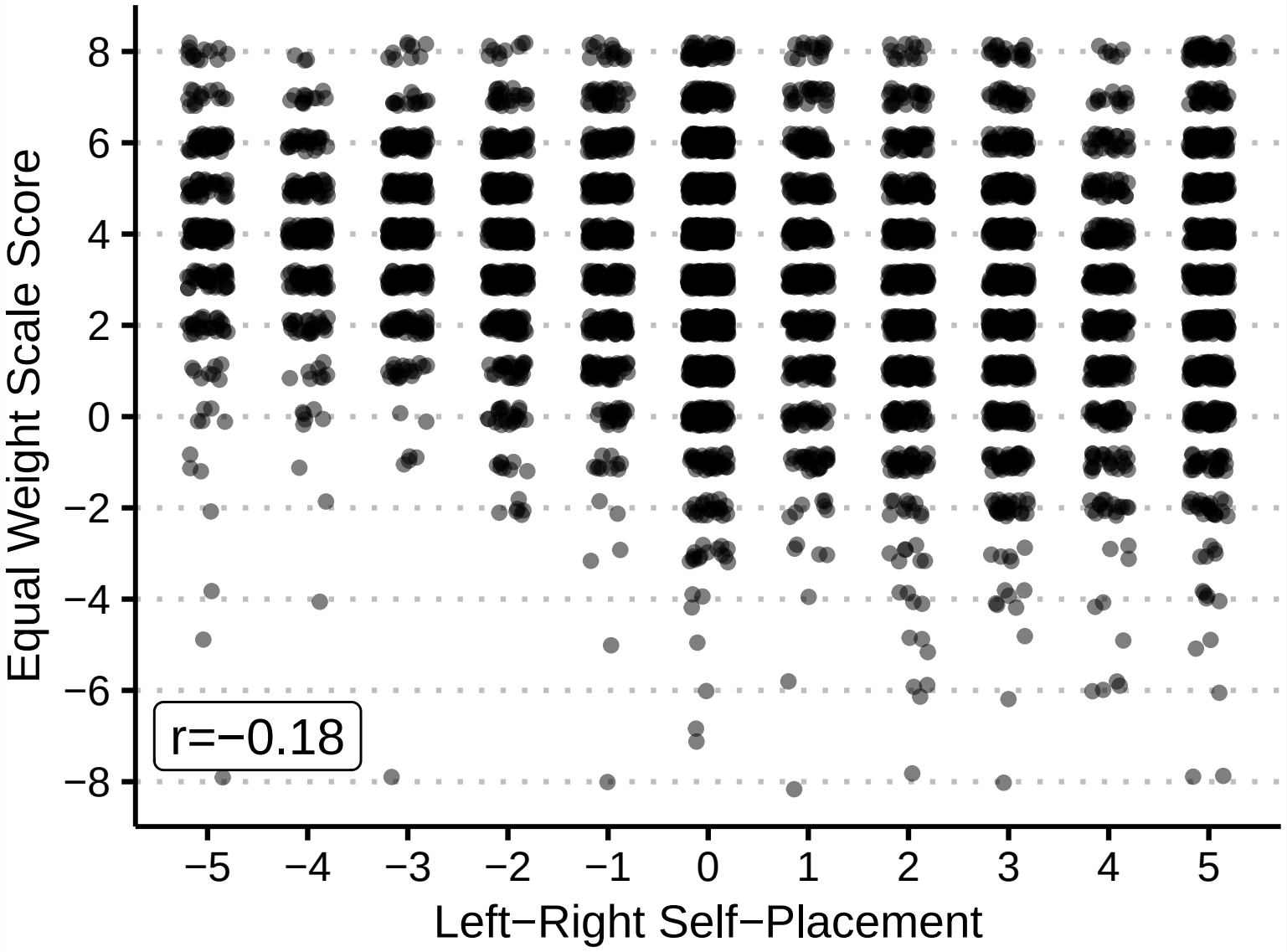
A simple, equal weighting approach

```
spending_battery$index <-
  rowSums(spending_battery[, grepl("*I_", names(spending_battery))])
```

- Code responses
  - Decrease (-1)
  - Keep the same (0)
  - Increase (+1)
- Aggregate additively
- Point scale running from -8 (less spending) to +8 (more spending)



Association with left-right self placement



Code for Figure

```
library(ggplot2)
library(ggthemes)

x <- round(cor(spending_battery$lr,spending_battery$total,
               use = "complete.obs"),2)

ggplot(spending_battery,
       aes(x= jitter(lr),y=jitter(index))) +
  geom_point(alpha=.5,size=1) +
  annotate("label",x=-4.5,y=-7,label=paste0("r=",x),size=4) +
  scale_y_continuous("Equal Weight Scale Score",breaks = seq(-8,8,by=2))+
  scale_x_continuous("Left-Right Self-Placement",breaks = -5:5) +
  theme_clean() +
  theme(axis.title = element_text(size=10),
        plot.background = element_rect(color=NA))
```

Association with vote choice

	Voted Trump	
	(1)	(2)
Equal Weight Scale	−0.320***	
	(0.014)	
I_Social_Security		−0.174**
		(0.083)
I_Public_Schools		−0.581***
		(0.084)
I_Border_Security		1.890***
		(0.077)
I_Crime		0.654***
		(0.083)
I_Welfare		−0.777***
		(0.072)
I_Highways		−0.244***
		(0.084)
I_Aid_Poor		−0.541***
		(0.080)
I_Environmental_Protection		−1.636***
		(0.083)
Constant	0.739***	0.296***
	(0.052)	(0.095)
Observations	5,653	5,653
Log Likelihood	−3,548.868	−1,721.743
Note:		*p<0.1; **p<0.05; ***p<0.01

Code for Table

```
library(stargazer)

glm_fit_1 <- glm(trump ~ total,
                 family=binomial(), data=spending_battery)
glm_fit_3 <- glm(trump ~ I_Social_Security + I_Public_Schools + I_Border_Security +
                   I_Crime + I_Welfare + I_Highways + I_Aid_Poor +
                   I_Environmental_Protection,
                 family=binomial(), data=spending_battery)

stargazer(glm_fit_1,glm_fit_3,
          header=F,
          keep.stat = c("n","ll"),
          dep.var.caption = "",
          dep.var.labels = c("Voted Trump"),
          covariate.labels = c("Equal Weight Scale"),
          no.space = T)
```

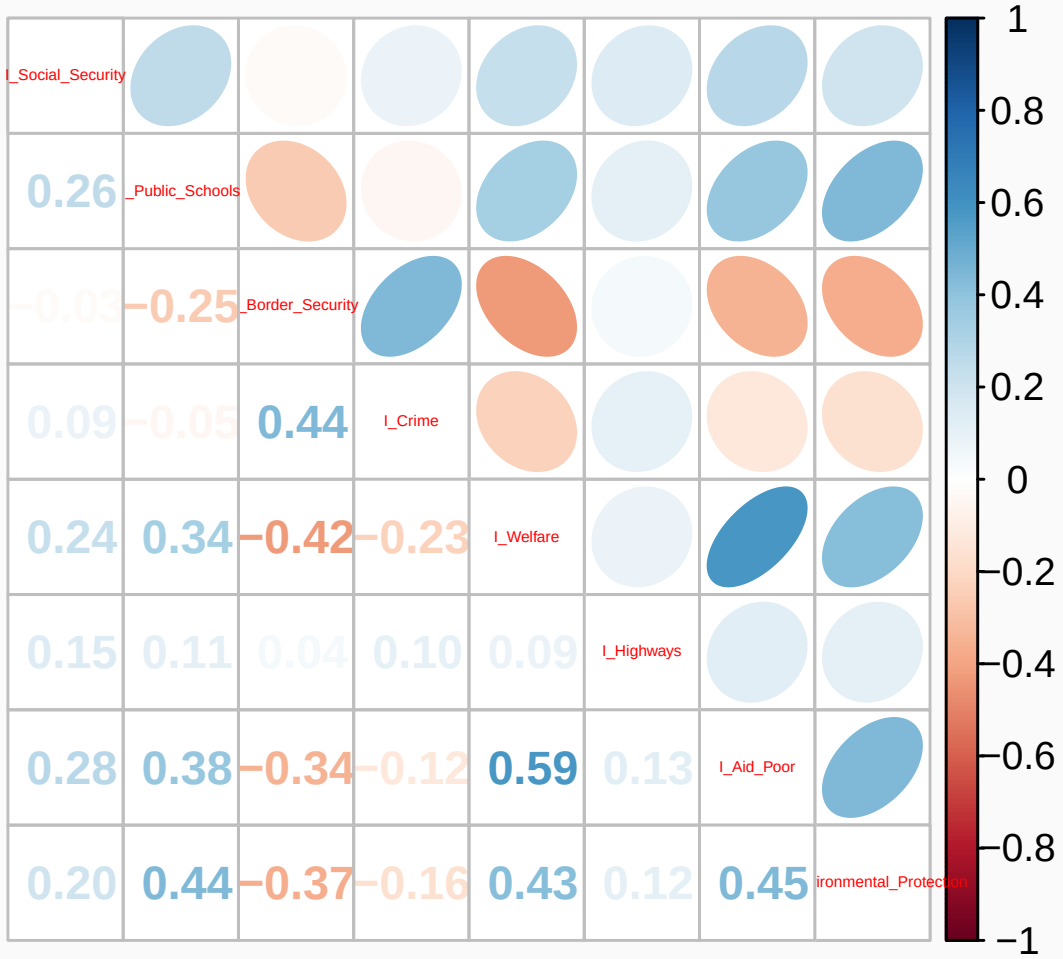
Limitations of the additive, equal weight scale

- Equal weight scale treats all spending domains as equally important
  - Social Security is 21% of US federal spending
  - Environmental Protection is less than 1% of US federal spending
- Equal weight commits us to the idea that "attitudes towards government spending" means more spending versus less spending.
- But what if by "attitudes towards government spending" we meant that some people like certain kinds of spending and other people like different kinds of spending?

Letting the data tell us how indicators go together

```
library(corrplot)

corrplot.mixed(
  cor(spending_battery[,1:8]),
  upper="ellipse",tl.cex=0.3)
```



- Which responses tend to go together in the data that we have?
- Does this *correlation* structure reveal anything useful?

Correlation

- None of these items are very strongly correlated with one another.
  - Strongest correlation is 0.59 between spending on welfare and aid to poor.
  - Strongest negative correlation is -0.42 between welfare and border security
  - Weakest correlation is 0.03 between social security and border security
  - Spending on highways is overall the least correlated with other items

Some broader patterns emerge

- No pairwise correlation is very high, but some indicators seem to go together more than others.
- Is there a good way to summarise which attitudes ‘go together’?

Principal Component Analysis (PCA)

Principal Component Analysis

- PCA aims to "decompose" a set of indicators variables into a series of "principal components"...
  - each of which is **uncorrelated** with all the other principal components
  - each of which only seeks to explain the **variation unexplained by the previous** principal components
- The principal components are ordered, from the most predictive to the least, **until there is no variation left to explain**
  - Using all of the principal components, you can reconstruct the original data exactly.

Mathematical details

- We have $(j = 1, 2, \ldots, p)$ observed variables/indicators I_{ij} measured for each unit i in a sample of data.
- Where $var\left(I_j\right)$ is the variance of observed variable j across all units i in the data, then the **total variance** of the p variables is:

$$\sum_{j=1}^{p} var\left(I_j\right)$$

  - This is the overall variation between the units in the data across all the variables.

Linearly describing the total variance

Because we like things that are linear, the principal components $m_{ik}\ (k = 1, 2, \ldots, p)$ are defined as **linear combinations** of the original variables:

$$\begin{aligned} m_{i1} &= a_{11}I_{i1} + a_{21}I_{i2} + \ldots + a_{p1}I_{ip} \\ m_{i2} &= a_{12}I_{i1} + a_{22}I_{i2} + \ldots + a_{p2}I_{ip} \\ &\vdots \\ m_{ip} &= a_{1p}I_{i1} + a_{2p}I_{i2} + \ldots + a_{pp}I_{ip} \end{aligned}$$

- In other words, each component m_{ik} is a weighted sum of the original I's, where the a_{jk} are **weights** or **coefficients**.

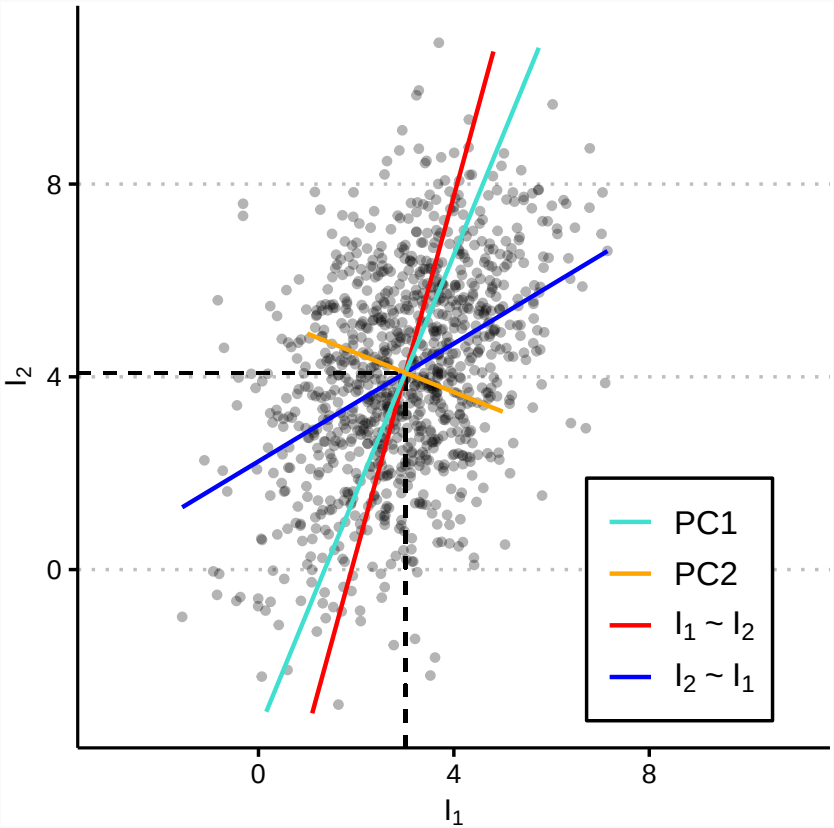
What are these components?

- Each component m_{ik} is a weighted sum of the original I's, where the a_{jk} are **weights** or **coefficients**.
- How do we select the a_{jk} so as to achieve the following goals?
  1. The new variables m_k are uncorrelated with one another
  2. They redescribe the total variation in the original x variables
  3. We describe as much of the variation as possible with the initial principal components.
- *The mathematical details about how we satisfy these three goals are not essential to understand.*
  - Some details are in the textbook, more in multivariate statistics textbooks linked from reading list.
  - Stuff about eigenvalues and eigenvectors, if you like that sort of thing.

Intuition of calculation

However, to understand *what* is calculated, it is worth understanding the contrast to linear regression.

- Imagine we have two correlated variables/indicators I_1 and I_2



- Regressing I_2 on I_1 minimises the sum of squares in the I_2 (vertical) direction
- Regressing I_1 on I_2 minimises the sum of squares in the I_1 (horizontal) direction
- Principal components minimise the *total* sum of squares in the diagonal directions

Properties of PCA

- PCA is sensitive to the scale of the original variables because it operates on the variances of those variables.
  - If you change the scale of *all* variables proportionately the principal components *will not change*.
  - If you change the scale of a *single* variable proportionately, the principal components *will change*.
- If you increase the variance of one of the variables but not the others, the initial principal components will increasingly reflect that variable rather than the others.
- This is particularly an issue if different variables have different units of measurement.

Properties of PCA

- Usually we do not want the results to be influenced by such differences and so the variables in a principal components analysis are typically **standardised** first.
  - Involves subtracting the sample mean from each observation and then dividing them by the sample standard deviation.
  - A standardised variable then has mean ? and standard deviation ?
- The total variance of p *standardized* variables is p, the number of variables.
- Since each variable contributes a variance of 1
  - PCA treats them all as having equal weight, and
  - puts equal weight on explaining variation in each variable.

When to standardise?

- If the variables are measured with different units, you almost always need to standardise first, or applying PCA makes no sense.
  - Following the same consistency of unit arguments that we talked about in week 3.
- Our example of the American National Election Study questions is one where there is an argument *against* standardisation.
  - The items are arguably already on a common scale: the decrease / keep the same / increase scale.

PCA in R

```r
pcafit <- prcomp(spending_battery[,1:8],scale=FALSE)

# pcafit$rotation
## provides the principal component coefficients

# pcafit$x
## provides the principal component values for original data

# pcafit$sdev
## provides the square roots of the eigenvalues
## (the variance explained by each principal component)
```

How many components?

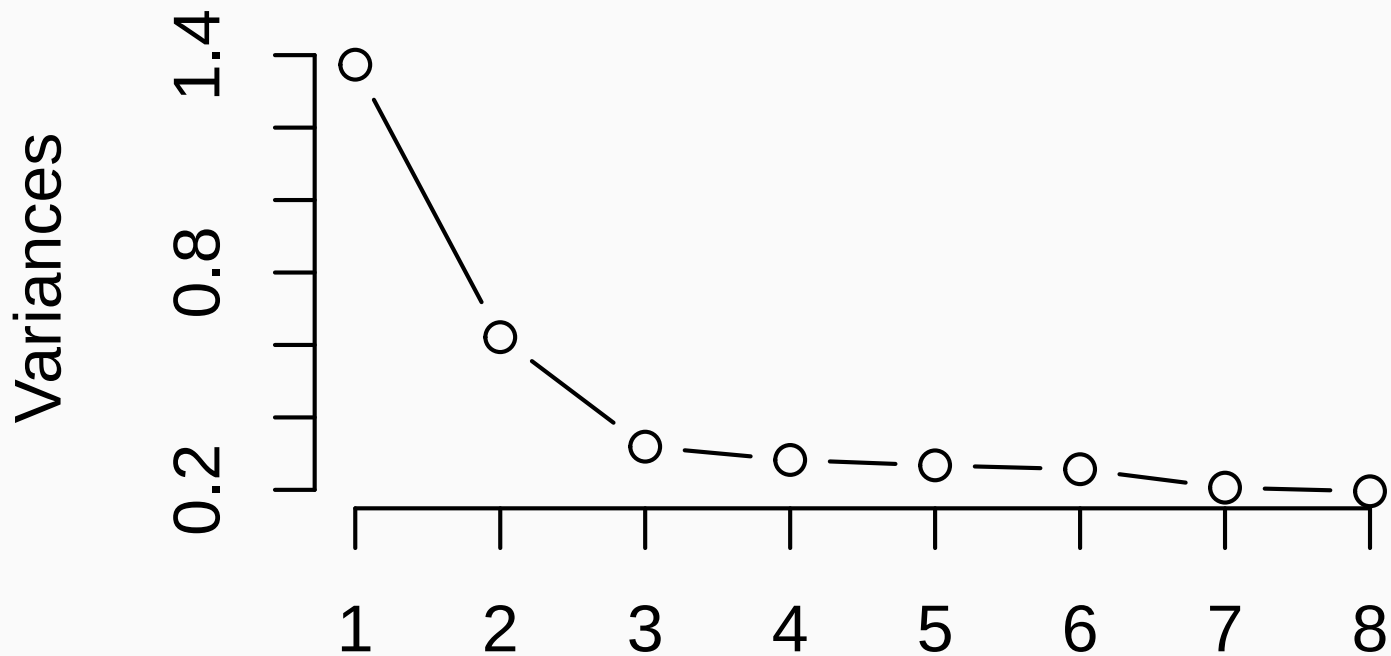
- On a data set with p variables, you will recover p principal components, in descending order of variance.
  - Put together, they describe **all** the variation in the data set.
  - The first component provides the best "single number" summary that is possible
  - The first two components the best "two number" summary, and so on.
- How many components are *enough* to describe the importance variation in a data set?
  1. **Absolute criteria**: components that explain some threshold of the total variation
  2. **Relative criteria**: components that have eigenvalues λ_k of at least some threshold
  3. **Relative criteria**: components that are upwards outliers in terms of variance explained

Screeplot

In practice, the third of these is most frequently used, because it is focuses on the components that most efficiently explain variation in the data set.

- This is often assessed visually using what is called a "screeplot".

```
screeplot(pcafit,type="lines",main="")
```



How many components?

- The first two principal components have substantially higher variance than any of the others.
  - They still 'only' explain 39% and 18% of the variation, respectively, but the third only accounts for 9%.
  - As with R^2 statistics, no strict criteria on how much variation explained is a lot.
  - With survey response scales like these, there is often a great deal of idiosyncratic variation.
- Major conclusion here: we should probably focus our attention on the content of the first two principal components in this instance.

Looking at the coefficients

Coefficients for first two principal components on 8 ANES spending questions.

```
two_component_table <- data.frame(
  PC1=round(pcafit$rotation[,"PC1"],2),
  PC2=round(pcafit$rotation[,"PC2"],2))
```

	PC1	PC2
I_Social_Security	0.14	-0.37
I_Public_Schools	0.29	-0.26
I_Border_Security	-0.50	-0.48
I_Crime	-0.26	-0.61
I_Welfare	0.52	-0.16
I_Highways	0.05	-0.27
I_Aid_Poor	0.42	-0.27
I_Environmental_Protection	0.37	-0.16

What to look for

- The **sign** of the coefficients tell you whether the principal component is positively or negatively correlated with responses to the item.
- The **magnitude** tells you whether the principal component is strongly or weakly correlated with responses to the item.

What are these dimensions?

$$
\begin{aligned}
m_{i1} &= a_{11}I_{i1} + a_{21}I_{i2} + ... + a_{p1}I_{ip} \\
m_{i2} &= a_{12}I_{i1} + a_{22}I_{i2} + ... + a_{p2}I_{ip} \\
&\vdots \\
m_{ip} &= a_{1p}I_{i1} + a_{2p}I_{i2} + ... + a_{pp}I_{ip}
\end{aligned}
$$

We can see by examining the signs of the a coefficients that:

- More positive values on PC1 are associated with support for increasing spending across most categories, except border security and criminal justice where the coefficients are negative
- More positive values on PC2 are associated with support for *decreasing* spending in *all* categories

Looking at the principal components

$$
\begin{aligned}
m_{i1} &= a_{11}I_{i1} + a_{21}I_{i2} + ... + a_{p1}I_{ip} \\
m_{i2} &= a_{12}I_{i1} + a_{22}I_{i2} + ... + a_{p2}I_{ip} \\
&\vdots \\
m_{ip} &= a_{1p}I_{i1} + a_{2p}I_{i2} + ... + a_{pp}I_{ip}
\end{aligned}
$$

- Let's now look at the left hand side, i.e. the principal components m_k, which are the **linear combinations** of the original variables
- Given the coefficients a and the observed responses I_{ij} in the data, we can construct the principal component values (or: scores) m_{ik} for every observation

Looking at the principal components

```r
# first PC
spending_battery$pc1 <- pcafit$x[,1]
summary(spending_battery$pc1)
```

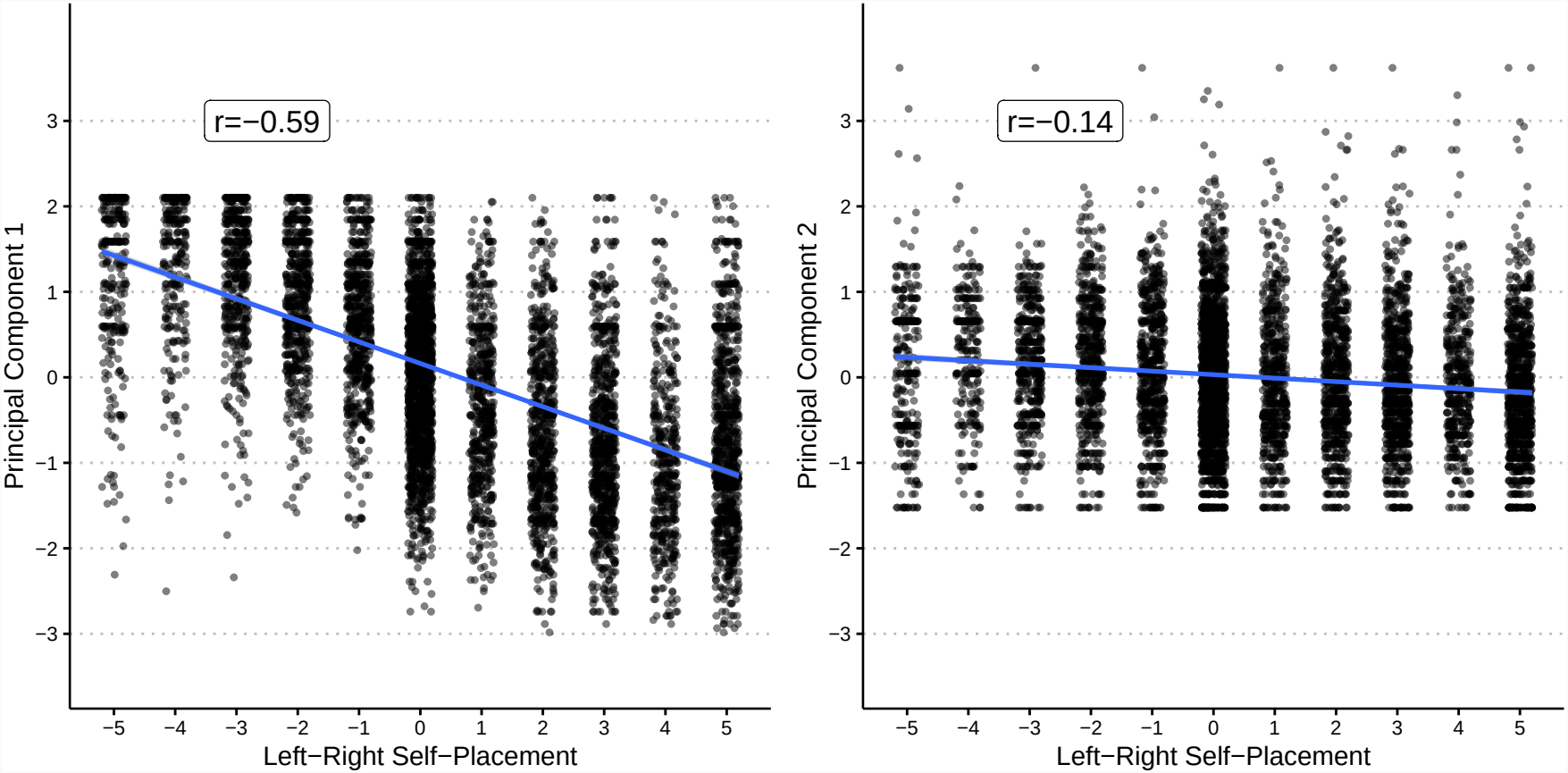
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.98339 -0.88831  0.05185  0.00000  0.89313  2.10068
```

```r
# second PC
spending_battery$pc2 <- pcafit$x[,2]
summary(spending_battery$pc2)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.52342 -0.56458 -0.03703  0.00000  0.56906  3.62036
```

This is the **measurement** we are interested in!

Associations with left-right self placement



- Both PCs are negatively associated with left-right position, but the first much more strongly than the secomd
- Suggests that our first PC might capture spending preferences that map onto the more salient US political divide, as respondents understand it

Associations with left-right self placement

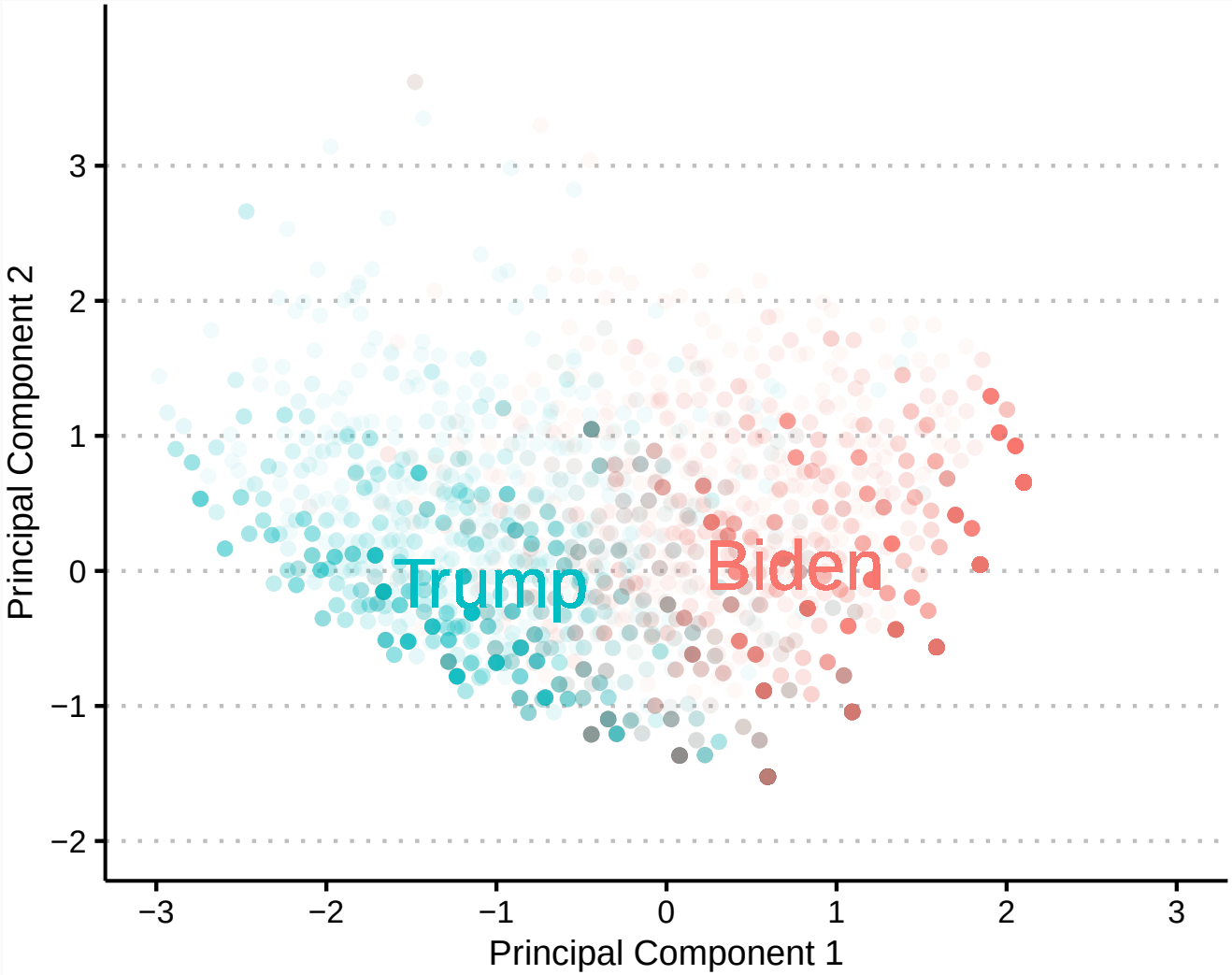
```
cor1 <- round(cor(spending_battery$lr, spending_battery$pc1,
                  use="pairwise.complete.obs"),2)
cor2 <- round(cor(spending_battery$lr, spending_battery$pc2,
                  use="pairwise.complete.obs"),2)

# Note: jitter() adds some random noise around a value - this is helpful otherwise
# everything would overlap in the plot
p1 <- ggplot(spending_battery, aes(x= jitter(lr),y=pc1)) +
  geom_point(alpha=.5, size=1) +
  geom_smooth(method = "lm") +
  annotate("label", x=-2.5, y=3, label=paste0("r=",cor1), size=5) +
  scale_y_continuous("Principal Component 1", breaks = seq(-3,3,by=1), limits = c(-3.5,4)) +
  scale_x_continuous("Left-Right Self-Placement", breaks = -5:5) +
  theme_clean()  +
  theme(axis.title = element_text(size=12),
        plot.background = element_rect(color=NA))

p2 <- ggplot(spending_battery, aes(x= jitter(lr),y=pc2)) +
  geom_point(alpha=.5, size=1) +
  geom_smooth(method = "lm") +
  annotate("label", x=-2.5, y=3, label=paste0("r=",cor2), size=5) +
  scale_y_continuous("Principal Component 2", breaks = seq(-3,3,by=1), limits = c(-3.5,4))  +
  scale_x_continuous("Left-Right Self-Placement", breaks = -5:5) +
  theme_clean() +
  theme(axis.title = element_text(size=12),
        plot.background = element_rect(color=NA))

ggpubr::ggarrange(p1,p2,ncol=2)
```

Associations with vote choice



- Trump and Biden voters differ much more on the first than on the second PC

Associations with vote choice

```r
library(tidyverse)
# The below is using some tidyverse code. don't worry too much about
# understanding it, this is just *in case* some of you are interested

tmp <- spending_battery[!is.na(spending_battery$vote20), ] %>% # remove missing vote2
  group_by(vote20) %>% # group by vote choice
  mutate(m1 = mean(pc1), m2 = mean(pc2)) %>% # mean of pc1 and pc2 by vote choice
  ungroup()  # ungroup data

ggplot(tmp,aes(x=pc1, y=pc2, color = vote20)) +
  geom_point(alpha=0.5) +
  geom_text(aes(x=m1,y=m2, label = vote20), size=7) +
  scale_y_continuous("Principal Component 2",
                     breaks = seq(-2,3,by=1), limits = c(-2,3.9)) +
  scale_x_continuous("Principal Component 1",
                     breaks = seq(-3,3,by=1), limits = c(-3,3)) +
  theme_clean() +
  theme(axis.title = element_text(size=10),
        plot.background = element_rect(color=NA),
        legend.position = "none")
```

Associations with vote choice

	Voted Trump		
	(1)	(2)	(3)
Equal Weight Scale	−0.320***		−0.232
	(0.014)		(0.146)
PC 1		−2.461***	−2.229***
		(0.064)	(0.158)
PC 2		−0.549***	−1.155***
		(0.054)	(0.385)
Constant	0.739***	−0.781***	−0.024
	(0.052)	(0.047)	(0.477)
Observations	5,653	5,653	5,653
Log Likelihood	−3,548.868	−1,827.985	−1,826.720
Note:			*p<0.1; **p<0.05; ***p<0.01

Associations with vote choice

```
glm_fit_1 <- glm(trump~total,
                 family=binomial(),data=spending_battery)
glm_fit_1b <- glm(trump~pc1+pc2,
                  family=binomial(),data=spending_battery)
glm_fit_2 <- glm(trump~total+pc1+pc2,
                 family=binomial(),data=spending_battery)

stargazer(glm_fit_1,glm_fit_1b,glm_fit_2,
          header=F,
          keep.stat = c("n","ll"),
          dep.var.caption = "",
          dep.var.labels = c("Voted Trump"),
          covariate.labels = c("Equal Weight Scale","PC 1","PC 2"),
          no.space = T)
```

PCA does not know left from right (or up from down)

$$
\begin{aligned}
m_{i1} &= a_{11}I_{i1} + a_{21}I_{i2} + ... + a_{p1}I_{ip} \\
m_{i2} &= a_{12}I_{i1} + a_{22}I_{i2} + ... + a_{p2}I_{ip} \\
&\vdots \\
m_{ip} &= a_{1p}I_{i1} + a_{2p}I_{i2} + ... + a_{pp}I_{ip}
\end{aligned}
$$

- If you multiply a_{11}, a_{21}, ..., and a_{p1} by -1:
  - the sign of the first principal component m_{i1} will flip for all observations
  - all the other principal components stay the same
  - the variances still add up to the right total
  - the principal components are still uncorrelated with one another
- You can always choose whichever sign is easier to talk about.

Unsupervised vs supervised learning

- Principal components are linear combinations of indicators just like the indices that we developed two weeks ago were linear combinations of indicators
  - The difference is in ***how*** we select the coefficients on the indicators.
- Previously, we considered several approaches to setting coefficients on indicators
  - Setting equal coefficients
  - Setting unequal coefficients based on theoretical/substantive considerations
  - Estimating coefficients from a training data set
  - Estimating coefficients from expert comparisons
- In all these cases, we provided **supervision** to the measurement problem to ensure that it measured what we wanted it to measure.

Unsupervised learning

- We ask the data which coefficients would best predict variation in the data set we were working with.
- We then look at the results to try to figure out what we measured.
- PCA is our first example of an **unsupervised** measurement method.
  - It is *unsupervised* in the sense that we have not indicated to the data what it is that we want to measure.
- Note though, even with methods like PCA that are usually described as unsupervised, *we still implicitly provide supervision through the choice of indicators that we include.*

Exploratory Factor Analysis (EFA)

Exploratory Factor Analysis

- Think back to week 4 when we talked about sports
  - **Discriminative measurement**: we could award points based on the results (eg 3-1-0 for win-draw-loss)
  - **Generative measurement**: we could fit a (Bradley-Terry) model where results arise because of latent strength
- We are going to do something similar here.
  - **Discriminative measurement**: Principal Component Analysis derives scales as linear combinations of observed indicators.
  - **Generative measurement**: Exploratory Factor Analysis hypothesizes latent dimensions, with the probability of observing different values of the observed indicators depending on where units are on those dimensions.

Mathematical details (O__o)

- We assume that each observation $i = 1, \dots, n$ has q latent factors

$$\theta_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{iq})$$

- We will assume that these latent factors $k = 1, \dots, q$ have a multivariate normal distribution:

$$\theta_i \sim N(\kappa, \phi)$$

  - mean vector κ (i.e. $E_i(\theta_k) = \kappa_k$ for $k = 1, \dots, q$)
  - covariance matrix
    - variances $var_i(\theta_k) = \phi_{kk}$ for $k = 1, \dots, q$ and
    - covariances $cov_i(\theta_k, \theta_{k'}) = \phi_{kk'}$ for $k, k' = 1, \dots, q;\ j \neq j'$

- We assume that the observed indicators I_{ij} for each observation i and each indicator $j = 1, \dots, p$ are related to the latent factors θ_{ik}:

$$I_{ij} = \alpha_j + \beta_{j1}\theta_{i1} + \beta_{j2}\theta_{i2} + \cdots + \beta_{jq}\theta_{iq} + \delta_{ij}$$

Yet another linear model

$$I_{ij} = \alpha_j + \beta_{j1}\theta_{i1} + \beta_{j2}\theta_{i2} + \cdots + \beta_{jq}\theta_{iq} + \delta_{ij}$$

- The model for each I_{ij} is a linear model depending on q factors.
- It looks a lot like PCA, but
  - ***Now the indicators are a linear function of the factors*** instead of the principal components being a linear function of the indicators.
  - **There is now an "error term"** δ_{ij} for each unit for each indicator at the end.
    - We assume that $\delta_{ij} \sim N(0, \sigma_j)$, and that these error terms are uncorrelated with each other, and uncorrelated with all the latent factor θ's as well.

Estimation & properties

- Factor analysis models are estimated in similar ways than regression models
  - The principle[1] is that we find the values of the latent factors θ and the loadings β that make it most likely that we would have observed the data that we did in fact observe.
- It is common to assume that the factors have mean $\kappa_k = 0$ and variance ϕ_{kk} = 1
  - This is because the latent factors could just as easily run from -1 to 1 or -100 to 100 or 0 to 10.
- The number of factors to be estimated is **decided by the analyst** ex-ante and there have to be *less* factors than there are indicators ($\neq$ PCA)
- The latent factor 1 could just as easily be factor 2 and vice versa and they can be correlated with each other ($\neq$ PCA)

[1] **Not** principal!

EFA in R

- Keep in mind that there are (infinitely) many equivalent solutions with different loadings – it depends on which 'rotation' is adopted
- We are going to use R's default `factanal()` implementation of factor analysis, which uses a "varimax" rotation that tends to yield results similar to PCA.

```
fafit <- factanal(spending_battery[,1:8],factors=2,
         rotation = "varimax",
         scores="regression")

# fafit$loadings
## provides the coefficients (loadings) of the factors for each indicator

# fafit$scores
## provides the estimated scores for each observation of each factor
```

Factor loadings

```
two_factor_table <- data.frame(F1=round(fafit$loadings[,1],2),
                               F2=round(fafit$loadings[,2],2))

kable(two_factor_table,booktabs=T,linesep="") %>%
  kable_styling(position = "center")
```

	F1	F2
I_Social_Security	0.44	0.13
I_Public_Schools	0.54	-0.12
I_Border_Security	-0.28	0.69
I_Crime	0.03	0.65
I_Welfare	0.64	-0.36
I_Highways	0.24	0.16
I_Aid_Poor	0.72	-0.22
I_Environmental_Protection	0.57	-0.27

Factor loadings vs PC coefficients

```
combined <- cbind(two_component_table, two_factor_table)[,c(1,3,2,4)]
```

	PC1	F1	PC2	F2
I_Social_Security	0.14	0.44	-0.37	0.13
I_Public_Schools	0.29	0.54	-0.26	-0.12
I_Border_Security	-0.50	-0.28	-0.48	0.69
I_Crime	-0.26	0.03	-0.61	0.65
I_Welfare	0.52	0.64	-0.16	-0.36
I_Highways	0.05	0.24	-0.27	0.16
I_Aid_Poor	0.42	0.72	-0.27	-0.22
I_Environmental_Protection	0.37	0.57	-0.16	-0.27

Factor loadings vs PC coefficients

- The patterns of loadings for the first dimension look broadly similar to the first principal component coefficients we saw earlier on the same data.
  - All spending categories have coefficients in the same direction except for crime, which has a coefficient close to zero.
  - The strongest coefficients are on welfare, child care and aid to poor.
  - The sign is flipped (positive instead of negative coefficients) but recall that this is entirely arbitrary.
- The second factor is quite a bit different than the second principal component.

Factor scores

```r
# First Factor
spending_battery$fa1 <- fafit$scores[,1]
summary(spending_battery$fa1)
```

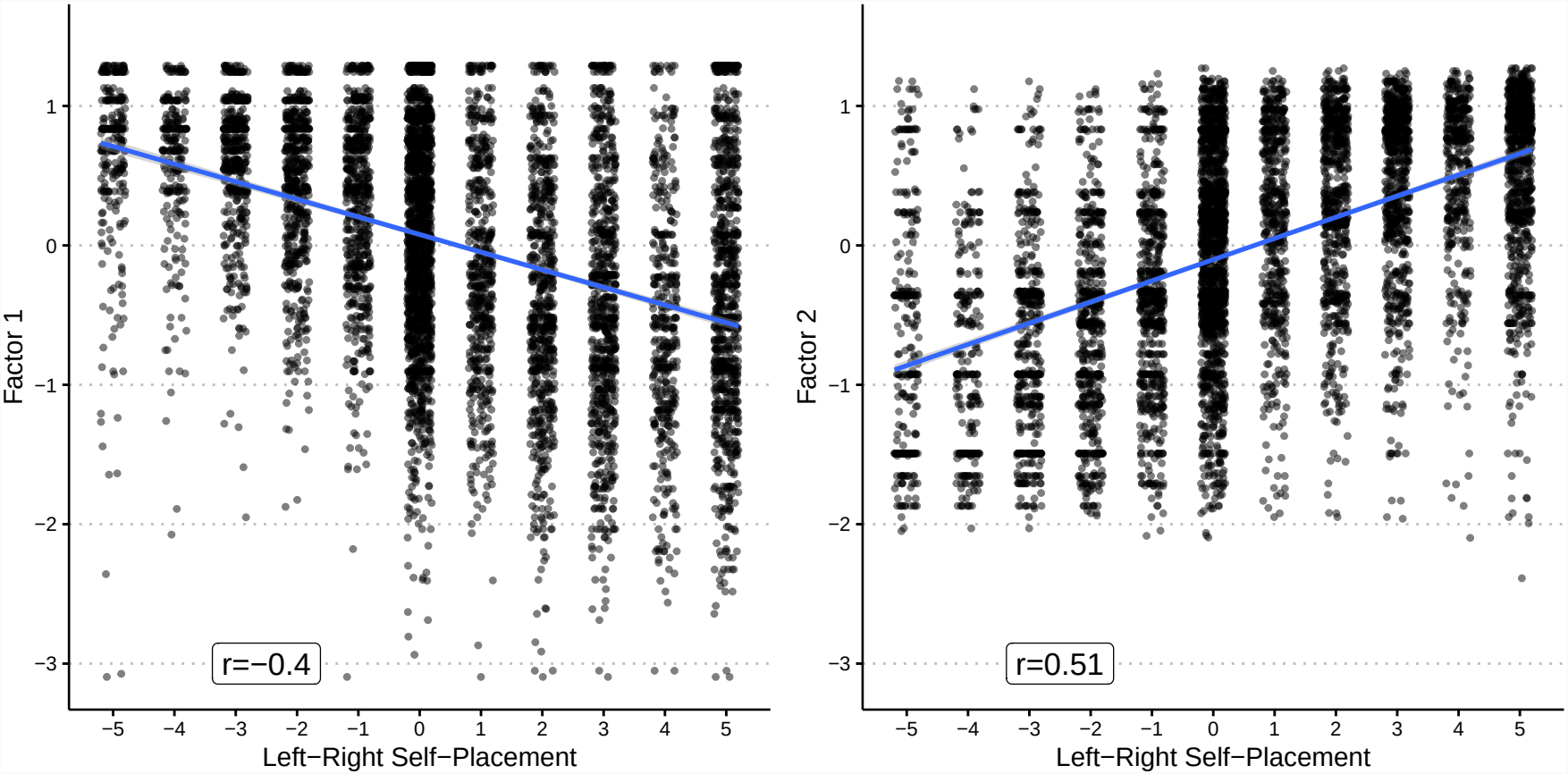
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.09547 -0.58634   0.07578   0.00000   0.70675   1.28885
```

```r
# Second Factor
spending_battery$fa2 <- fafit$scores[,2]
summary(spending_battery$fa2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.3875 -0.5449  0.1479  0.0000  0.7621  1.2707
```

This is the **measurement** we are interested in!

Associations with left-right self placement



- The first factor is negatively and the second is positively associated with left-right placement, with the second marginally more strongly so
- Both seem to map on somewhat on what people think of as left-right placement, but in different ways

Associations with left-right self placement

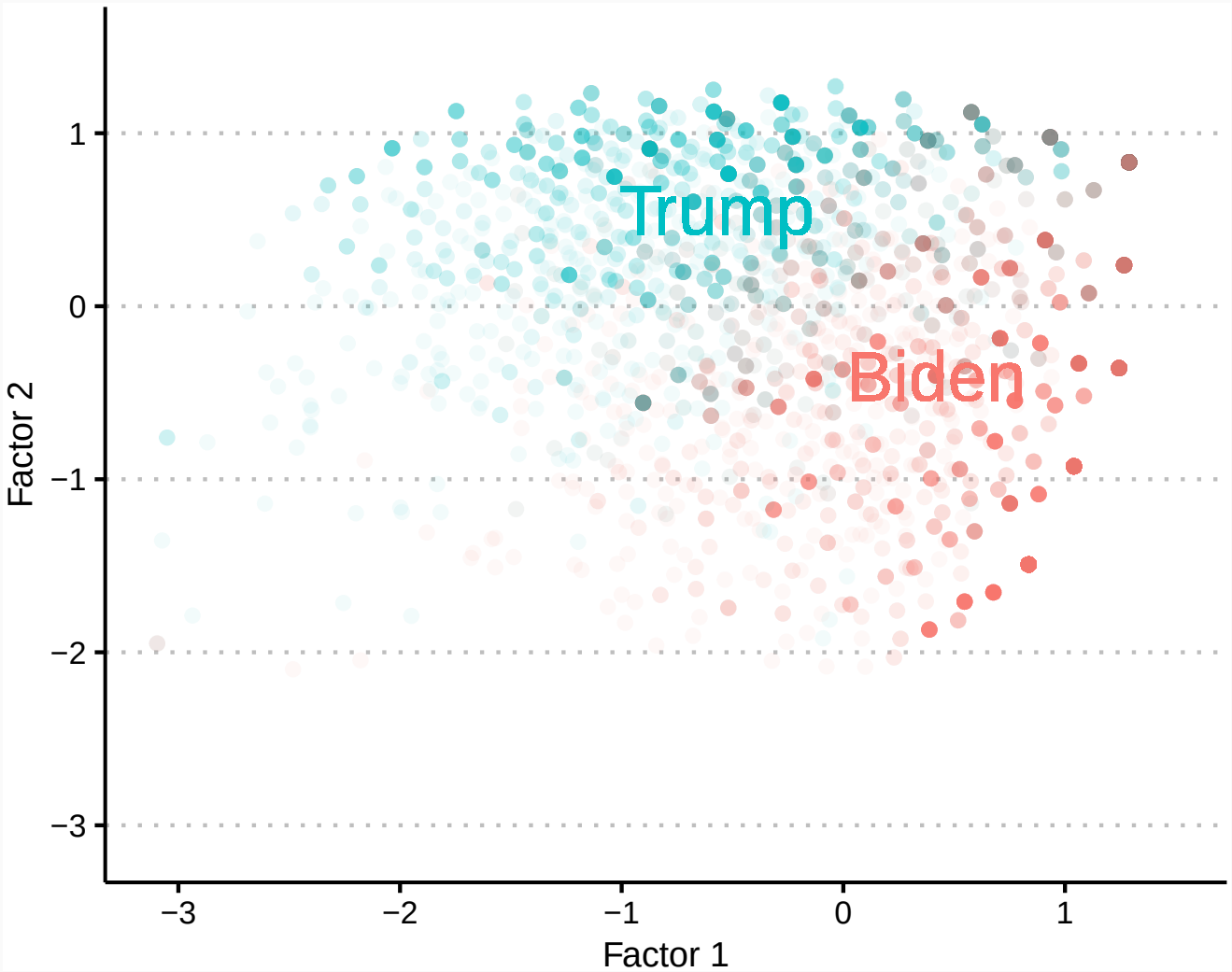
```
cor1 <- round(cor(spending_battery$lr, spending_battery$fa1,
                  use="pairwise.complete.obs"),2)
cor2 <- round(cor(spending_battery$lr, spending_battery$fa2,
                  use="pairwise.complete.obs"),2)

p1 <- ggplot(spending_battery, aes(x= jitter(lr),y=fa1)) +
  geom_point(alpha=.5, size=1) +
  geom_smooth(method = "lm") +
  annotate("label", x=-2.5, y=-3, label=paste0("r=",cor1), size=5) +
  scale_y_continuous("Factor 1", breaks = seq(-3,1,by=1), limits = c(-3.1,1.5)) +
  scale_x_continuous("Left-Right Self-Placement", breaks = -5:5) +
  theme_clean() +
  theme(axis.title = element_text(size=12),
        plot.background = element_rect(color=NA))

p2 <- ggplot(spending_battery, aes(x= jitter(lr),y=fa2)) +
  geom_point(alpha=.5, size=1) +
  geom_smooth(method = "lm") +
  annotate("label", x=-2.5, y=-3, label=paste0("r=",cor2), size=5) +
  scale_y_continuous("Factor 2", breaks = seq(-3,1,by=1), limits = c(-3.1,1.5))  +
  scale_x_continuous("Left-Right Self-Placement", breaks = -5:5) +
  theme_clean() +
  theme(axis.title = element_text(size=12),
        plot.background = element_rect(color=NA))

ggpubr::ggarrange(p1,p2,ncol=2)
```

Associations with vote choice



- Trump and Biden voters seem to be different along the dimensions of both factors

Associations with vote choice

```
tmp <- spending_battery[!is.na(spending_battery$vote20),] %>%
  group_by(vote20) %>%
  mutate(f1 = mean(fa1), f2 = mean(fa2)) %>%
  ungroup()

ggplot(tmp,aes(x=fa1, y=fa2, color = vote20)) +
  geom_point(alpha=.05) +
  geom_text(aes(x=f1,y=f2, label = vote20), size=7) +
  scale_y_continuous("Factor 2",breaks = seq(-3,1,by=1), limits = c(-3.1,1.5)) +
  scale_x_continuous("Factor 1", breaks = seq(-3,1,by=1), limits = c(-3.1,1.5)) +
  theme_clean() +
  theme(axis.title = element_text(size=10),
        plot.background = element_rect(color=NA),
        legend.position = "none")
```

Factor scores vs PC values

Pairwise correlations between scores from PCA (first two components) and EFA (two factor model, varimax rotation):

```
pc_fa_cor_table <-
  cor(as.matrix(spending_battery[,c("pc1","pc2","fa1","fa2")]))
```

	pc1	pc2	fa1	fa2
pc1	1.00	0.00	0.82	-0.70
pc2	0.00	1.00	-0.56	-0.72
fa1	0.82	-0.56	1.00	-0.16
fa2	-0.70	-0.72	-0.16	1.00

- Looking at the factor scores estimated for each survey respondent i, we see that the first dimension is highly correlated with the first principal component, but then things get a bit more complicated.

PCA vs EFA

- PCA and EFA are more conceptually different than practically different.
    1. PCA summarises variance in the indicators as efficiently as possible in terms of components that are linear functions of the indicators.
    2. EFA is an effort to identify the latent factors that would have been most likely to generate the indicators, if in fact the indicators were generated by latent factors according to a specificed linear model.
- In the end, both trying to provide a “simple” summary of the correlations across variables in the data.
    - When they are applied to the same data, they tend to yield similar conclusions in the first dimension at least.

Analogy to linear regression

- Linear regression can be motivated in two ways.
  1. As a simple “summary” transformation of the data: the linear projection of the observed data onto the explanatory variables that minimises the sum of square errors ($\longrightarrow$ discriminative measurement)
  2. As the best estimate of a linear “generative” model with normally distributed errors ($\longrightarrow$ generative measurement)
- For linear regression these two motivations give identical answers, for PCA/EFA we get approximately the same answers.

Discriminative measures versus generative measures

- We often have a choice between
  1. an approach based on the logic of summarising a data set in a simple way
  2. an approach based on the looking at the estimated parameters from a model that could have generated the observed data
- What is different?
  - **Discriminative** approaches tend to be simpler to implement, faster to compute, and clearly limited in their interpretation.
  - **Generative** approaches provide a direct way of describing uncertainty about measures, while also being more demanding to think about, more computationally demanding, and to risk over-interpretation.

From covariation to interpretation

Over-interpretation

- Factor analysis models tempt one to make causal claims that are not justified
  - Same as Bradley-Terry models, if you really "believe" in the unidimensional strength model
  - Same as linear regression models, if you really "believe" in the linear model
- It is better to view the model as a summary of the variation in y conditional on x, without causal assumptions.
  - You do not need to "believe" in the factor to do factor analysis and for it to be *useful*
  - It is almost always a mistake to adopt a causal interpretation of the factors
  - The factors do not really exist just because you setup a model with factors in it

Do not “reify” the factors!

- What would reifying the factors mean in this context?
  - People are walking around with a general latent attitude towards government spending, and their responses to each item reflects this
  - Nothing in our analysis justifies this!
- Again, by analogy...
  - a non-zero β_j coefficient in a linear regression model does not imply a causal effect of I_j on y.
  - all you have demonstrated is that there is a partial association of I_j with y given a model with some set of other x variables.
- In our current case, the fact that a set of variables are correlated with one another does not mean that they are all the product of a single common latent factor.

Repeat until funny

What's the point?

- So, having made the point about what these techniques do not license you to say, what exactly are they good for?
  - They are good for exploring/summarising data sets
- Thinking back to two weeks ago:
  - Why wouldn't you always do it this way?
  - Why did we bother with the material in the previous weeks?
  - Why ever use equal or expert-specified weights when you can measure the weights from the data?

Do we know what we are measuring?

- PCA/EFA measure what explains the most variation in the data, *not what best represents any concept we might be interested in*.
- We have no direct control over what PCA/EFA are measuring
  - The dimensions/factors that explain variation best maybe the concept we wanted to measure.
  - Or they may be a mixture of what we want to measure and something else.
  - Or they may be other things entirely.
- We have indirect control over what the methods are measuring because we determined the data that went into the estimation.

How simple a summary?

- An important issue for all unsupervised measurement: how many components/factors/dimensions do want?
- Statistical criteria tell you about how much variation in the data you can describe with different numbers of components/factors/dimensions.
  - Screeplot for PCA, and analogous measures of relative fit for EFA models with different number of factors.
- Statistical criteria may still not tell you whether the components/factors/dimensions that you have recovered are useful or interpretable!

Summing up

- Instead of deciding (based on previous knowledge) how our indicators should be aggregated, we could just let the data tell us
- **Unsupervised measurement** describes a series of methods that allow us to recover measures based on the *patterns* in the data
- In PCA, each component is a linear function of the indicator variables, and the coefficients describe the association between each indicator and the respective PC
- in EFA, each indicator variable is a linear function of the factor(s), and the coefficients describe the association between each factor and the respective indicator
- Taking an unsupervised approach requires a lot of post-hoc interpretation to make sense of the results and you can never be sure that you are measuring what you intended!