SocViz

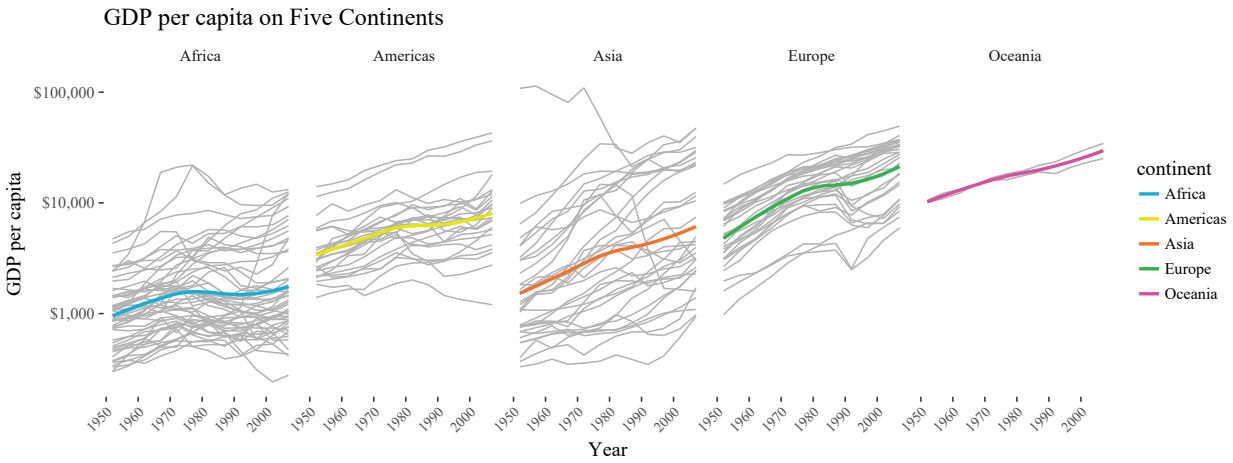
```
library(socviz)
library(ggplot2)
library(ggthemes)
library(dplyr)
library(gapminder)
library(extrafont)
theme_set(theme_tufte(base_size = 14))
ubdc_palette <- c("#13AFD6", "#E6E600", "#F07329", "#35B14E", "#D7509A", "#2165AF",
                  "#BCD032","#866BAC", "#545A5D",  "#7A8082", "#E2D988", "#628DB7",
                  "#929B9A", "#93B8DA", "#31649B", "#FBF8D0", "#ACB2B4")
```

Chapter 4.3

Facets and small multiples

Also shows `geom_smooth`

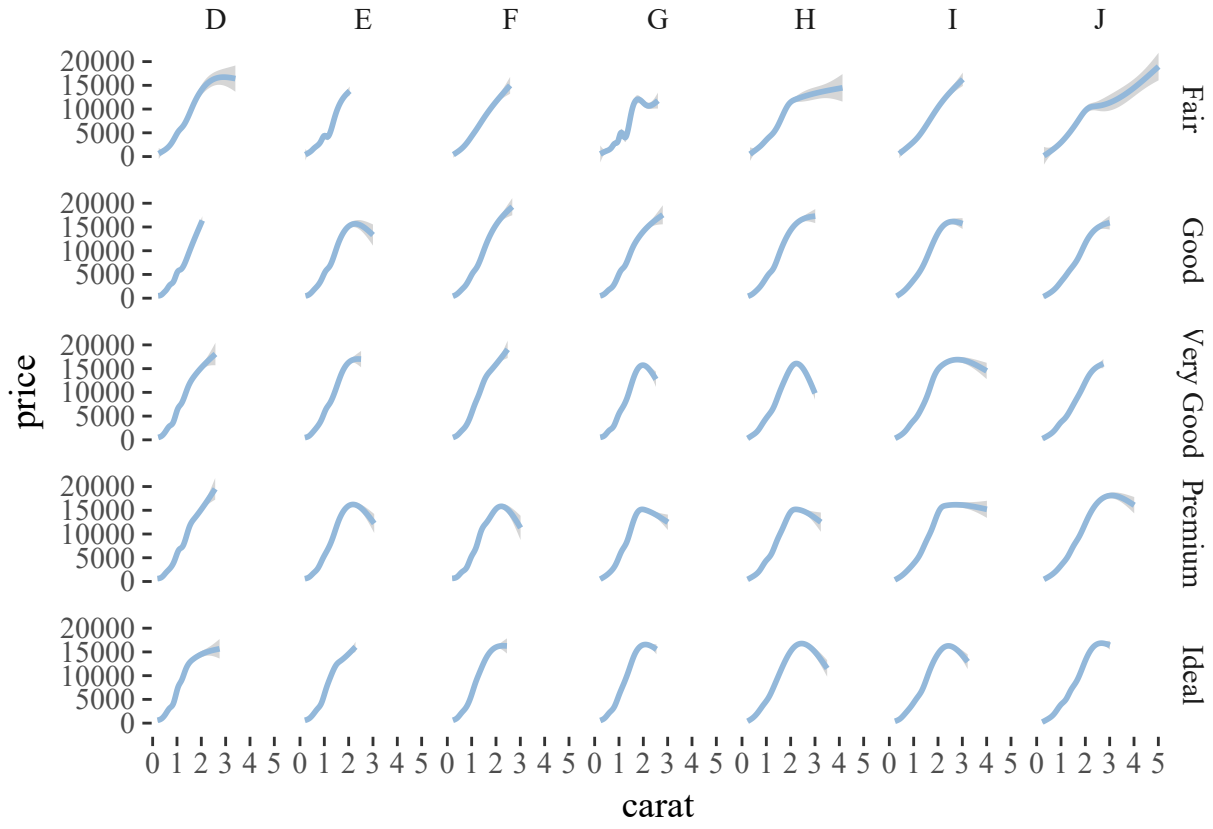
```
p <- ggplot(data = gapminder, mapping = aes(x = year, y = gdpPercap, color = continent))

p +
  geom_line(color="gray70", aes(group = country)) +
  geom_smooth(size = 1.1, method = "loess", se = FALSE) +
  scale_y_log10(labels=scales::dollar) +
  theme(axis.text.x = element_text(angle = 45, size = 10, hjust = 1, vjust = 1)) +
  facet_wrap(~ continent, ncol = 5) +
  labs(x = "Year",
       y = "GDP per capita",
       title = "GDP per capita on Five Continents") +
  scale_color_manual(values = ubdc_palette)
```



The `facet_wrap()` function is best used when you want a series of small multiples based on a single categorical variable. Your panels will be laid out in order and then wrapped into a grid. If you wish you can specify the number or rows or the number of columns in the resulting layout. Facets can be more complex than this. For instance, you might want to cross-classify some data by two categorical variables. In that case you should try

`facet_grid()` instead. This function will lay out your plot in a true two-dimensional arrangement, instead of a series of panels wrapped into a grid.

```
p <- ggplot(data = diamonds, mapping = aes(x = carat, y = price))

p + geom_smooth(color = ubdc_palette[14]) +
  facet_grid(cut ~ color)  #or (~ cut + color) or (cut ~ color + clarity)
```



Proportional bar charts (with extrafonts added)

Need to summarise data first!!

```
rel_by_region <-
  gss_sm %>%  #from socviz package
  group_by(bigregion, religion) %>%
  summarize(N = n()) %>%
  mutate(freq = N / sum(N),
         pct = round((freq*100), 1))
rel_by_region
```

bigregion	religion	N	freq	pct
Northeast	Protestant	158	0.3237705	32.4
Northeast	Catholic	162	0.3319672	33.2
Northeast	Jewish	27	0.0553279	5.5
Northeast	None	112	0.2295082	23.0
Northeast	Other	28	0.0573770	5.7

bigregion	religion	N	freq	pct
Northeast	NA	1	0.0020492	0.2
Midwest	Protestant	325	0.4676259	46.8
Midwest	Catholic	172	0.2474820	24.7
Midwest	Jewish	3	0.0043165	0.4
Midwest	None	157	0.2258993	22.6
Midwest	Other	33	0.0474820	4.7
Midwest	NA	5	0.0071942	0.7
South	Protestant	650	0.6178707	61.8
South	Catholic	160	0.1520913	15.2
South	Jewish	11	0.0104563	1.0
South	None	170	0.1615970	16.2
South	Other	50	0.0475285	4.8
South	NA	11	0.0104563	1.0
West	Protestant	238	0.3765823	37.7
West	Catholic	155	0.2452532	24.5
West	Jewish	10	0.0158228	1.6
West	None	180	0.2848101	28.5
West	Other	48	0.0759494	7.6
West	NA	1	0.0015823	0.2

Sanity check

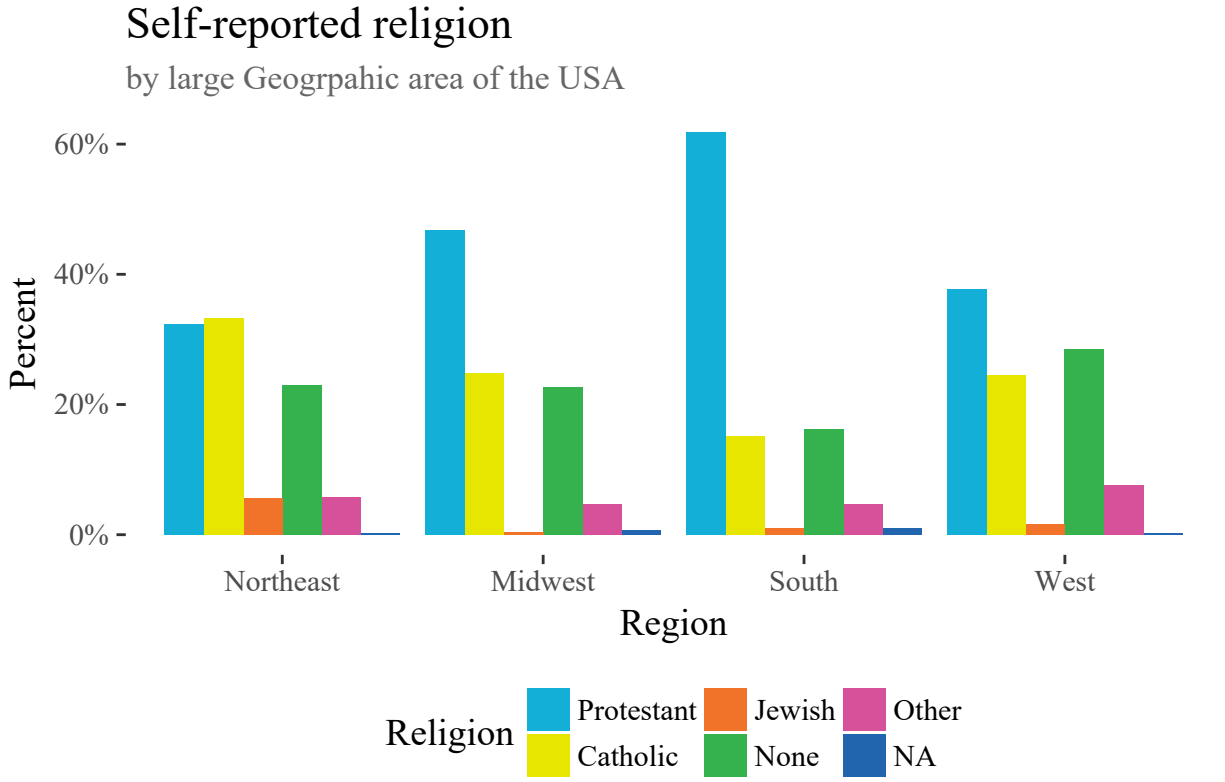
```
rel_by_region %>%
  group_by(bigregion) %>%
  summarize(total = sum(pct))
```

bigregion	total
Northeast	100.0
Midwest	99.9
South	100.0
West	100.1

Rounding errors only, now plot

```
p <- ggplot(rel_by_region,
            aes(
              x = bigregion,
              y = freq,          #I have changed this from percent and using scale_y_percent below
              fill = religion))

ubdc_colours <- p +
  geom_bar(position = "dodge", stat = "identity") +
  labs(x = "Region",
       y = "Percent",
       fill = "Religion",
       title = "Self-reported religion",
       subtitle = "by large Geogrpahic area of the USA",
       caption = "based on plot created in Data Visualization for Social Science") +
  scale_y_continuous(labels = scales::percent) + #Using freq instead of percent variable
  theme(legend.position = "bottom",
        plot.subtitle = element_text(color="#666666"),
```

```
        plot.caption = element_text(color="#AAAAAA", size=10)) +
  scale_fill_manual(values = ubdc_palette, na.value = ubdc_palette[6])
ubdc_colours
```



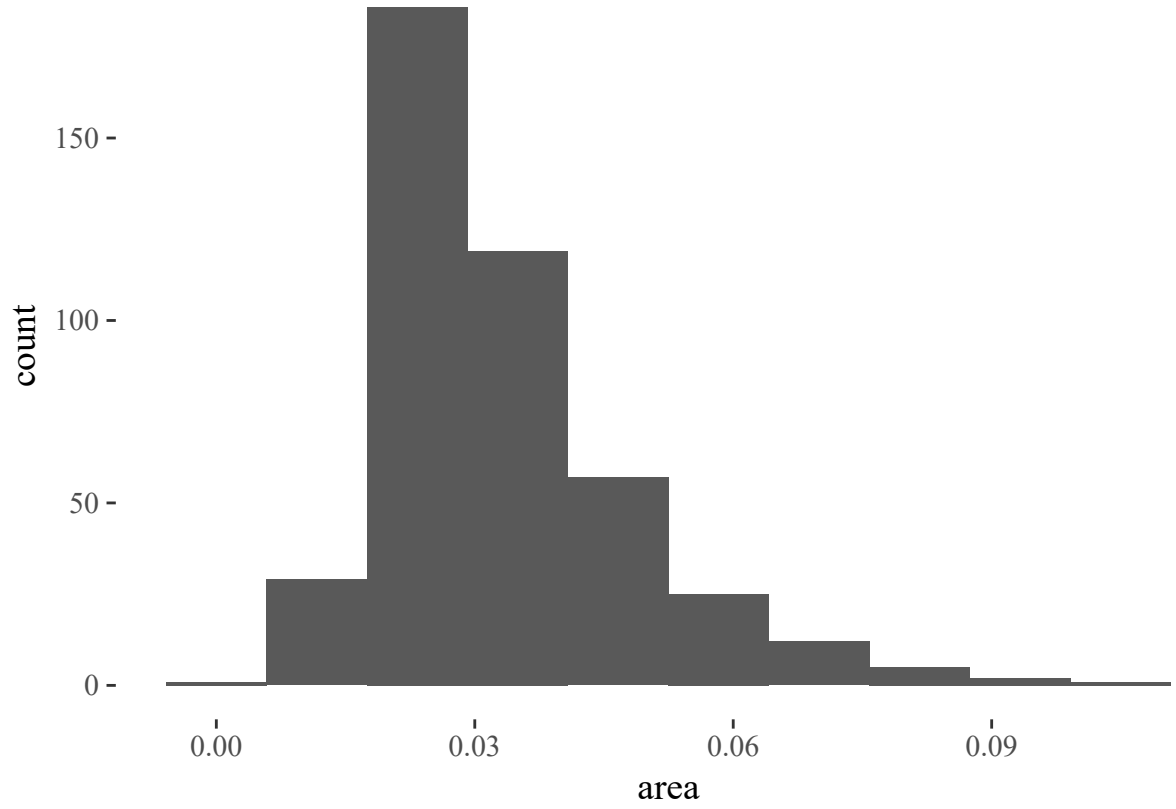
4. 7 Histograms and Freqpoly

```
midwest <- midwest
midwest %>%
  slice(1:10) %>%
  select(1:8) #for pdf viewing
```

PID	county	state	area	poptotal	popdensity	popwhite	popblack
561	ADAMS	IL	0.052	66090	1270.9615	63917	1702
562	ALEXANDER	IL	0.014	10626	759.0000	7054	3496
563	BOND	IL	0.022	14991	681.4091	14477	429
564	BOONE	IL	0.017	30806	1812.1176	29344	127
565	BROWN	IL	0.018	5836	324.2222	5264	547
566	BUREAU	IL	0.050	35688	713.7600	35157	50
567	CALHOUN	IL	0.017	5322	313.0588	5298	1
568	CARROLL	IL	0.027	16805	622.4074	16519	111
569	CASS	IL	0.024	13437	559.8750	13384	16
570	CHAMPAIGN	IL	0.058	173025	2983.1897	146506	16559

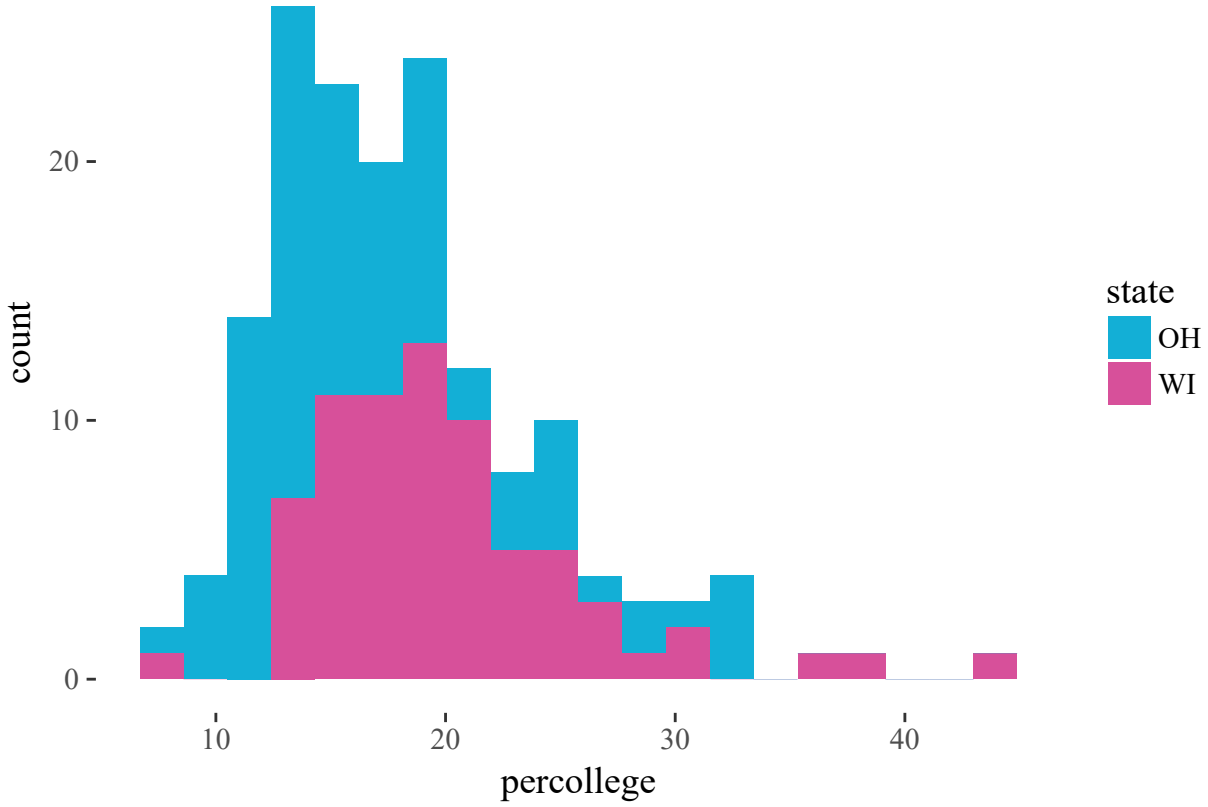
Basic histogram

```
ggplot(midwest, aes(area)) +
  geom_histogram(bins = 10)
```



Grouped histogram (with selected palette colours)

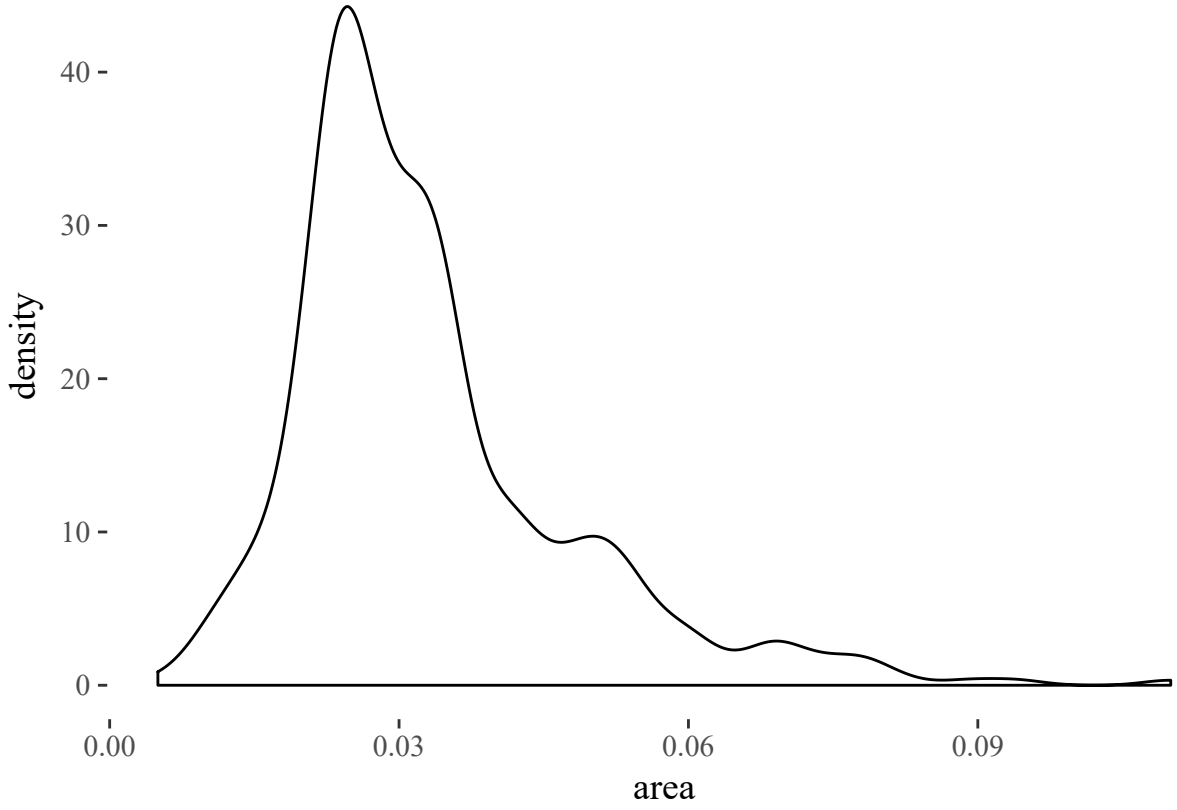
```
midwest %>%
  filter(state %in% c("OH", "WI")) %>%
  ggplot(aes(percollege, fill = state)) +
  geom_histogram(bins = 20) +
  scale_fill_manual(values = ubdc_palette[c(1, 5)])
```



Kernal density plots

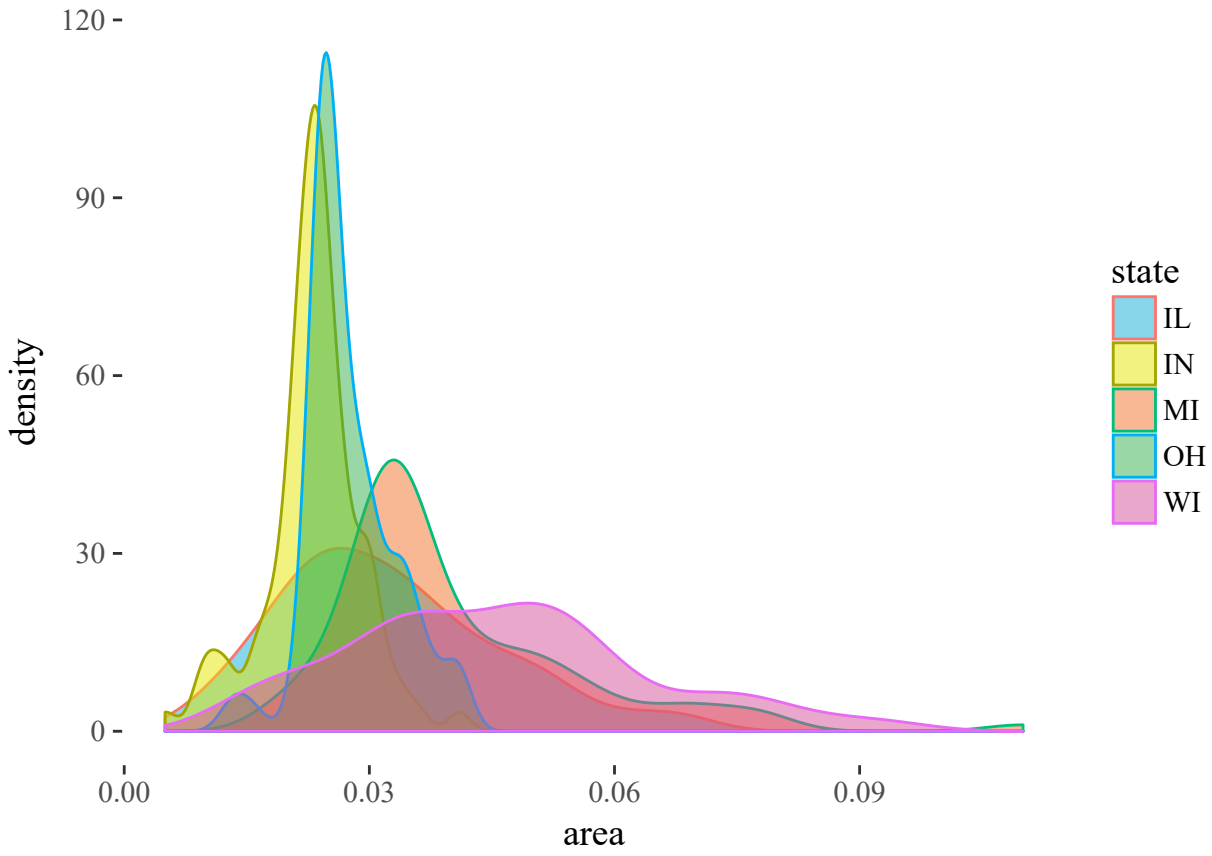
Basic plot

```
ggplot(midwest, aes(area)) +
  geom_density()
```



Multi-factor plot

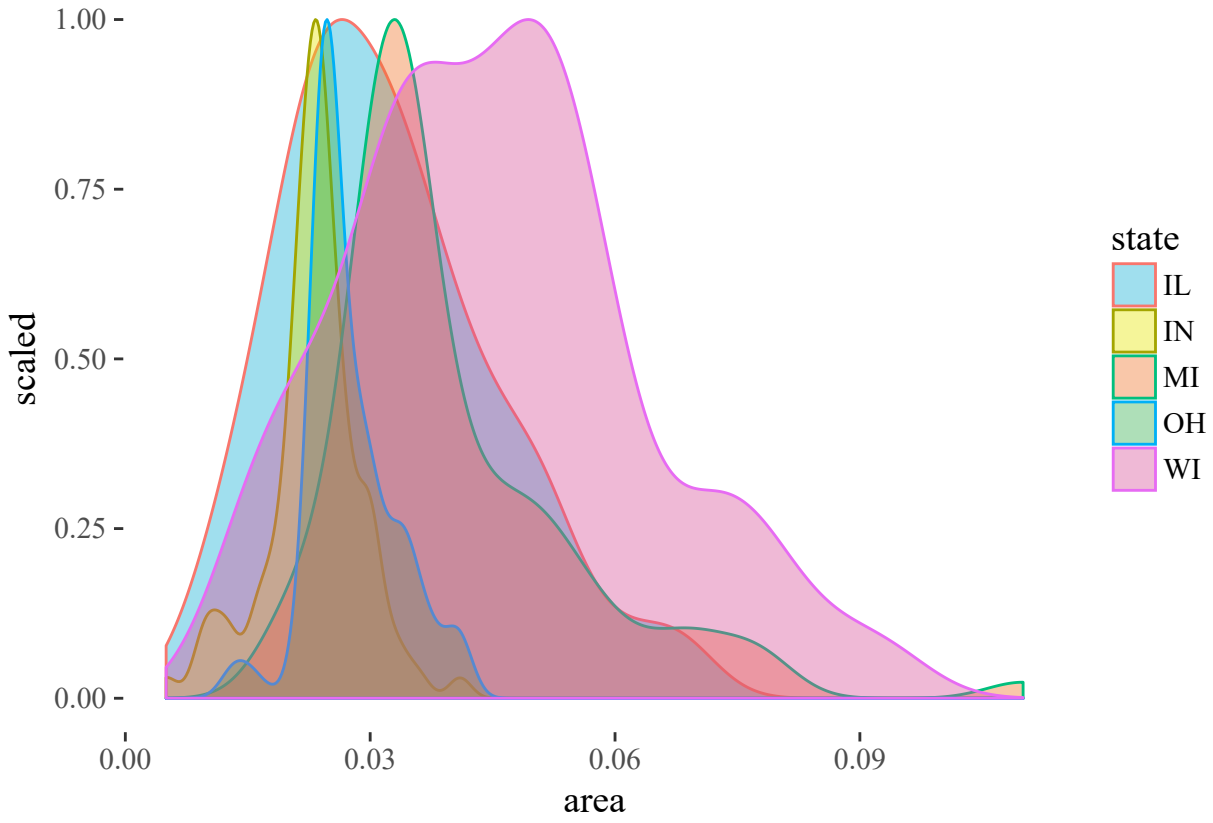
```
ggplot(midwest, aes(area, fill = state, color = state)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = ubdc_palette)
```




```
#alternative
# ... aes(area, color = state)) +
#     geom_line(stat = "density")
#
#helpful for busy plots like below
```

Proportional density estimate scaled to max 1

```
ggplot(midwest, aes(area, fill = state, color = state)) +
  geom_density(alpha = 0.4, aes(y = ..scaled..)) +
  scale_fill_manual(values = ubdc_palette)
```



tidy up

```
rm(list = c("midwest", "p", "rel_by_region", "ubdc_colours"))
```

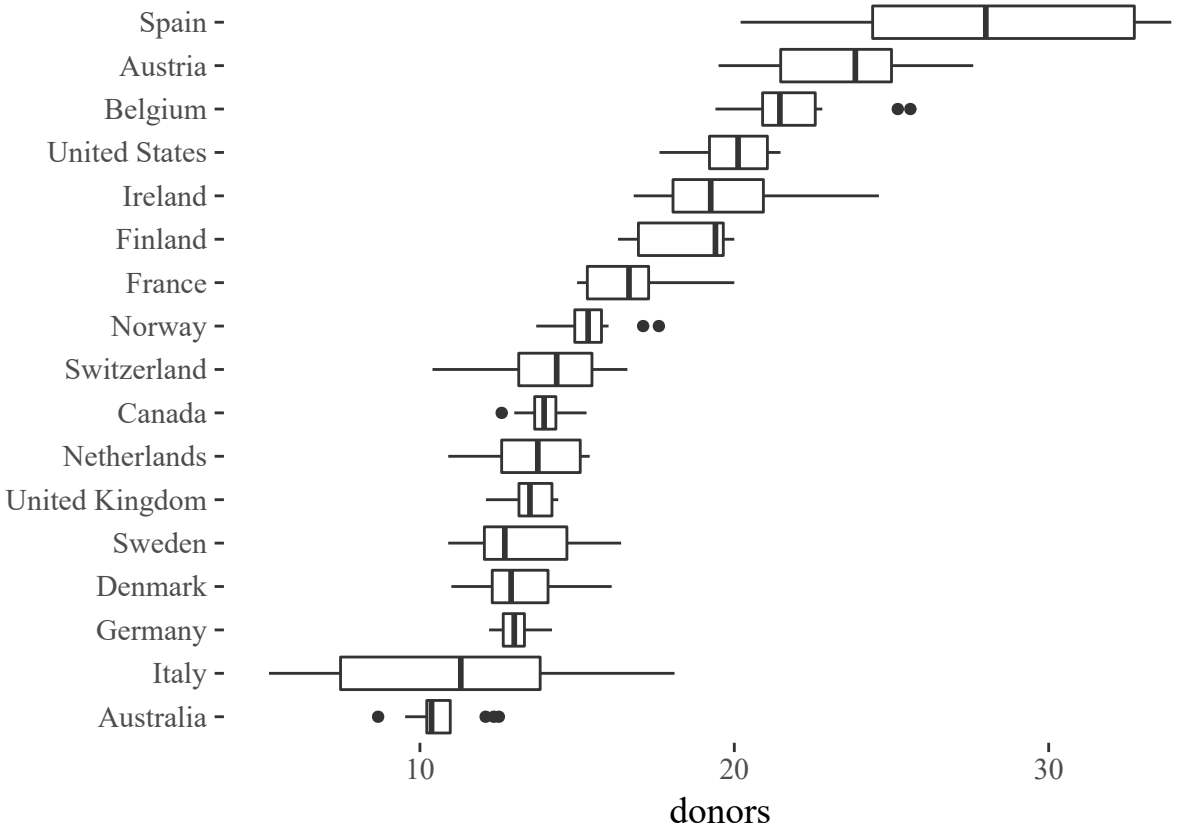
Chapter 5.1 Continous variables by group or category

```
organdata <- organdata
organdata %>%
  select(1:7) %>%
  head(., n = 10)
```

country	year	donors	pop	pop.dens	gdp	gdp.lag
Australia	NA	NA	17065	0.2204433	16774	16591
Australia	1991-01-01	12.09	17284	0.2232723	17171	16774
Australia	1992-01-01	12.35	17495	0.2259980	17914	17171
Australia	1993-01-01	12.51	17667	0.2282198	18883	17914
Australia	1994-01-01	10.25	17855	0.2306484	19849	18883
Australia	1995-01-01	10.18	18072	0.2334516	21079	19849
Australia	1996-01-01	10.59	18311	0.2365389	21923	21079
Australia	1997-01-01	10.26	18518	0.2392129	22961	21923
Australia	1998-01-01	10.48	18711	0.2417061	24148	22961
Australia	1999-01-01	8.67	18926	0.2444834	25445	24148

Ordered boxplot

```
p <- ggplot(organdata, aes(x = reorder(country, donors, na.rm = TRUE),
                           y = donors))
p + geom_boxplot() +
  labs(x = NULL) +
  coord_flip()
```

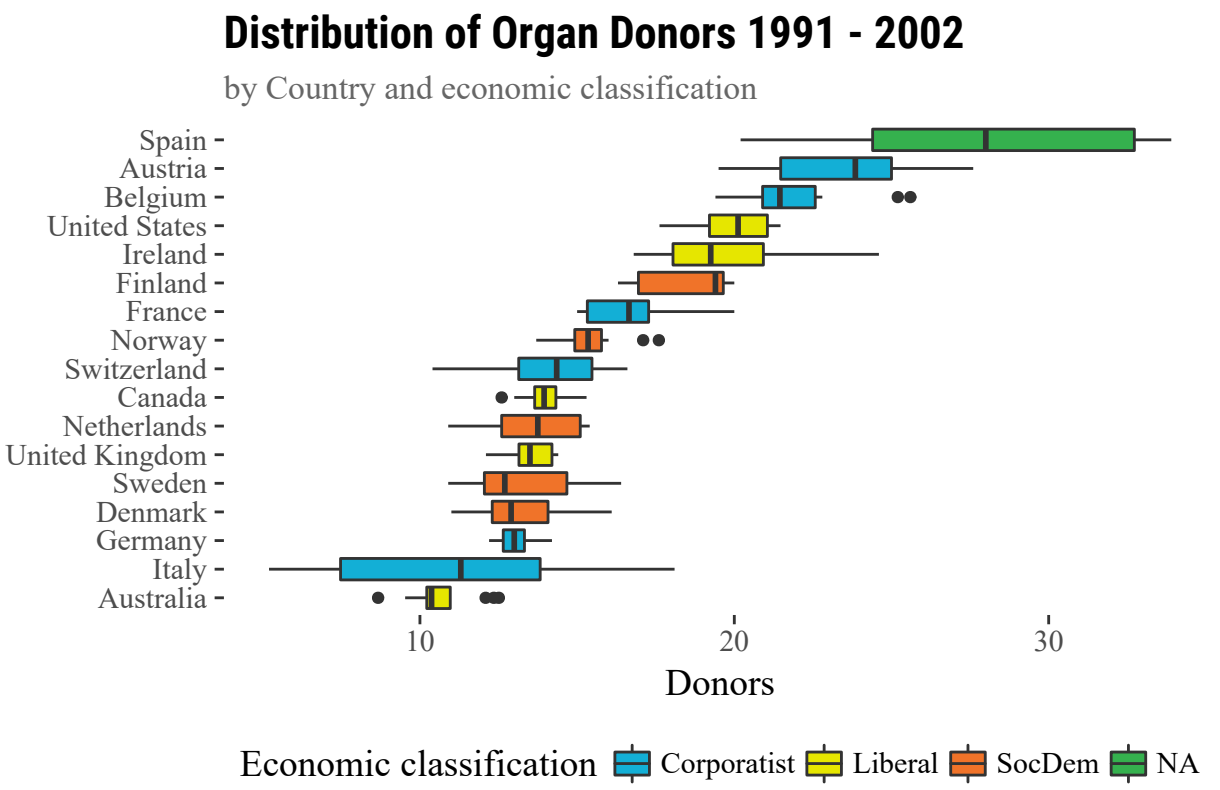


The reorder() function takes two required arguments. The first is the categorical variable or factor that we want to reorder. In this case, that's country. The second is the variable we want to reorder it by. Here that is the donation rate, donors. The third and optional argument to reorder() is the function you want to use as a summary statistic. By default, that is, if you only give reorder() the first two required arguments, it will reorder the categories of your first variable by the mean value of the second. You can name any sensible function you like to reorder the categorical variable (e.g., median, or sd).

Ordered boxplot filled by second categorical variable

```
p <- ggplot(organdata, aes(x = reorder(country, donors, na.rm = TRUE),
                           y = donors, fill = world))
p + geom_boxplot() +
  labs(x = NULL,
       y = "Donors",
       title = "Distribution of Organ Donors 1991 - 2002",
       subtitle = "by Country and economic classification",
       caption = "Using theme_hc()",
       fill = "Economic classification") +
```

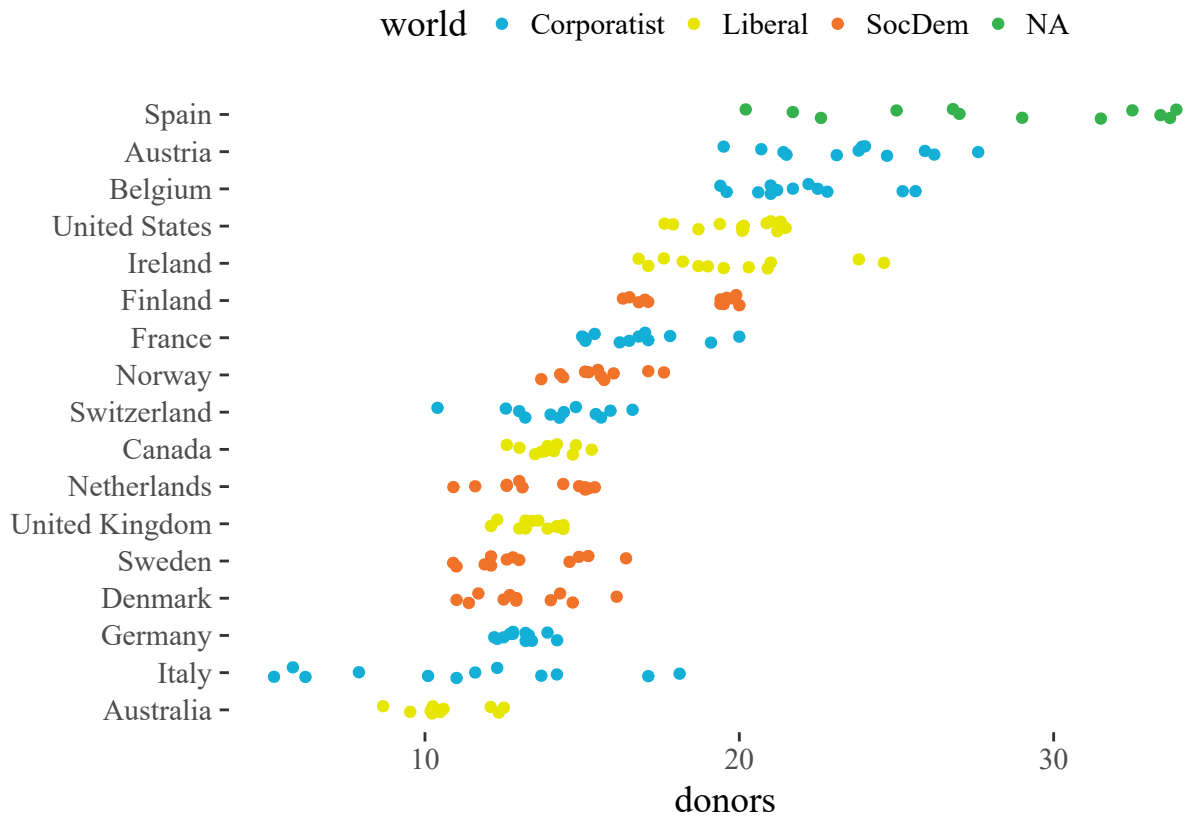
```
  coord_flip() +
  scale_fill_manual(values = ubdc_palette, na.value = ubdc_palette[4]) +
  theme(legend.position = "bottom",
        plot.subtitle = element_text(color="#666666"),
        plot.title = element_text(family="Roboto Condensed", face = "bold"),
        plot.caption = element_text(color="#AAAAAA", size=10))
```



Jitter

Same but using points (and jitter). Use "color" instead of "fill" for points. Useful when number of obsrvations within each category is small.

```
p <- ggplot(data = organdata,
            mapping = aes(x = reorder(country, donors, na.rm=TRUE),
                          y = donors, color = world))

p + geom_jitter(position = position_jitter(width=0.15)) +
  labs(x=NULL) +
  coord_flip() +
  scale_color_manual(values = ubdc_palette, na.value = ubdc_palette[4]) +
  theme(legend.position = "top")
```



Clevland dotplot

Good alternative to a ba chart (`stat = "identity"`, or `geom_col()`)

For catgorical variable with only one point per category.

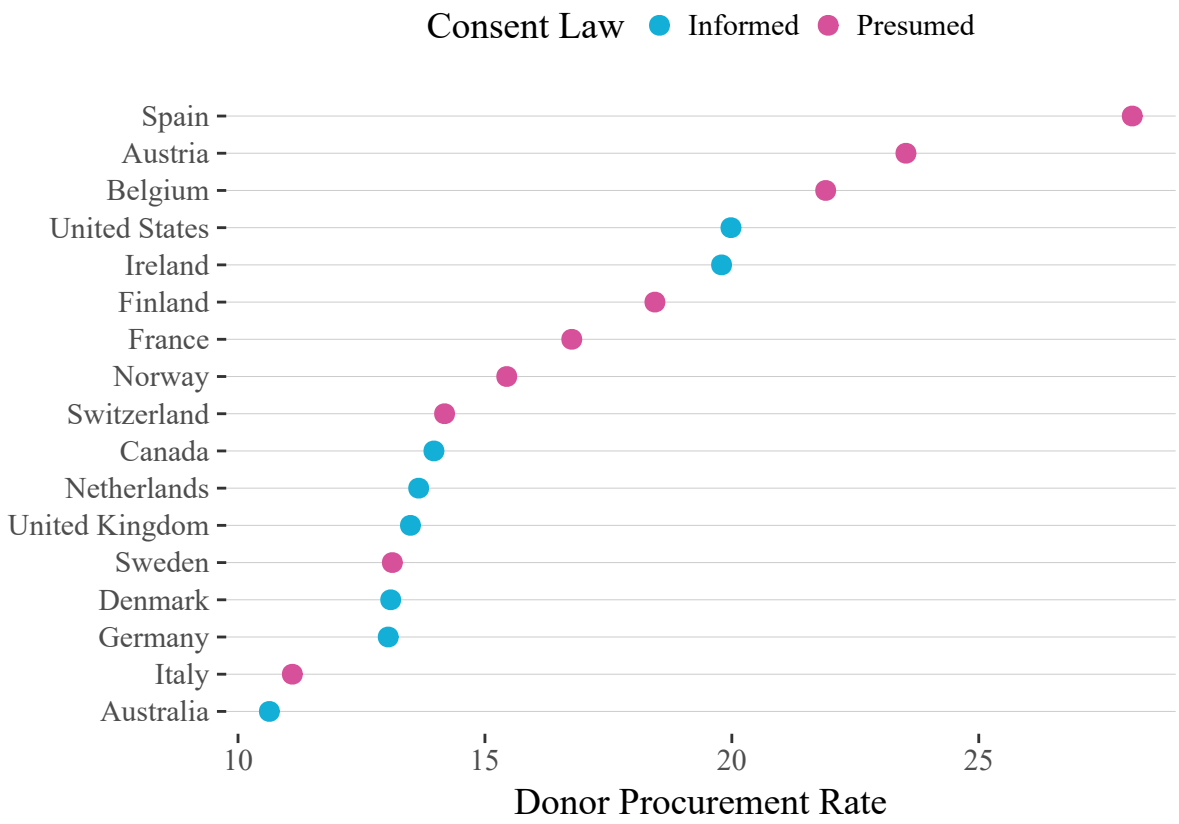
Need to pre-summarise data

```
by_country <- organdata %>% group_by(consent.law, country) %>%
    summarize(don.rate = mean(donors, na.rm = TRUE),
              don.sd = sd(donors, na.rm = TRUE),
              gdp = mean(gdp, na.rm = TRUE),
              health = mean(health, na.rm = TRUE),
              roads = mean(roads, na.rm = TRUE),
              cerebvas = mean(cerebvas, na.rm = TRUE))
by_country
```

consent.law	country	don.rate	don.sd	gdp	health	roads	cerebvas
Informed	Australia	10.63500	1.1428075	22178.54	1957.500	104.87573	557.6923
Informed	Canada	13.96667	0.7511607	23711.08	2271.929	109.26011	422.3846
Informed	Denmark	13.09167	1.4681208	23722.31	2054.071	101.63635	640.6923
Informed	Germany	13.04167	0.6111960	22163.23	2348.750	112.78873	706.7692
Informed	Ireland	19.79167	2.4784373	20824.38	1479.929	117.77424	704.6923
Informed	Netherlands	13.65833	1.5518074	23013.15	1992.786	76.09357	584.9231
Informed	United Kingdom	13.49167	0.7751344	21359.31	1561.214	67.92936	707.9231
Informed	United States	19.98167	1.3253667	29211.77	3988.286	155.16783	444.3846

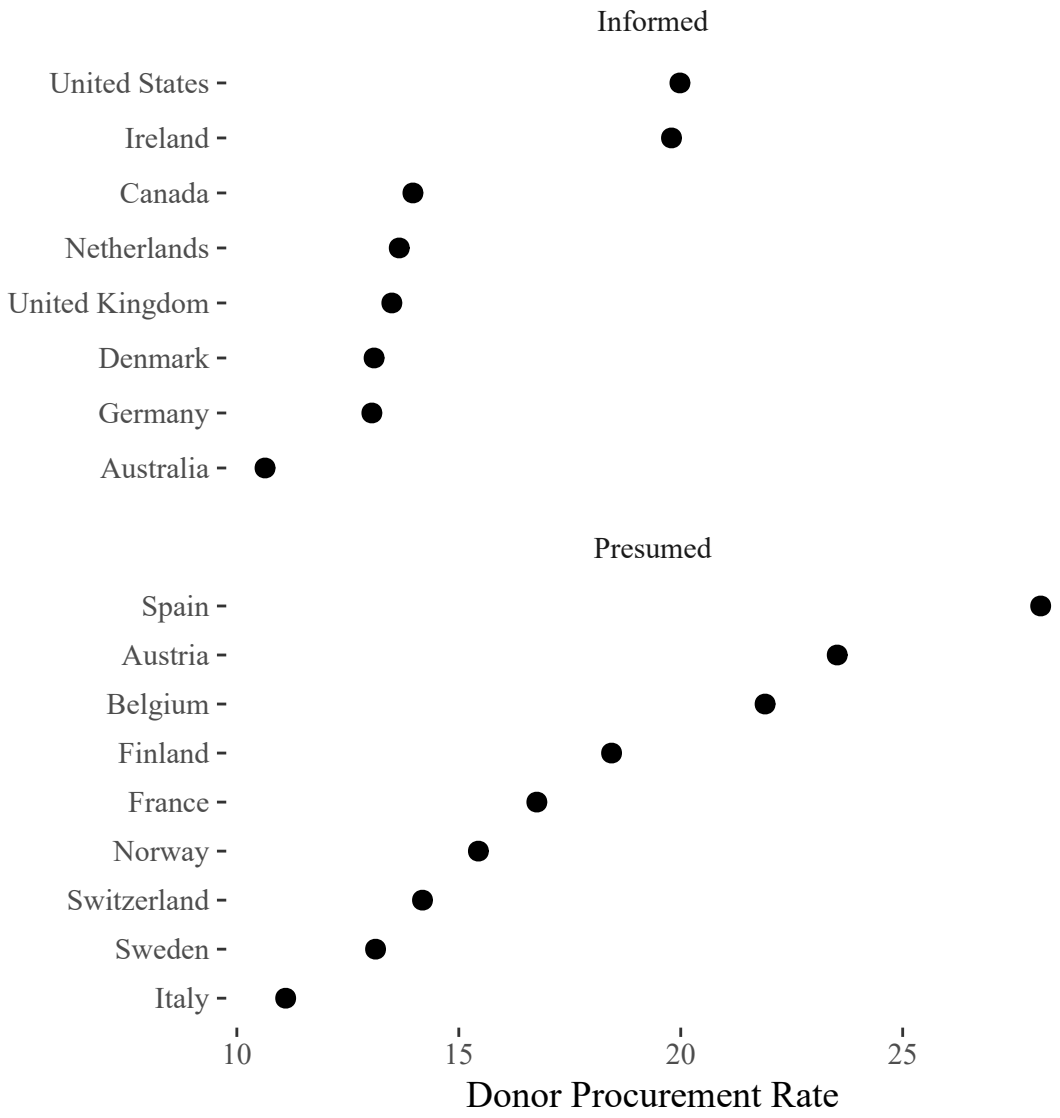
consent.law	country	don.rate	don.sd	gdp	health	roads	cerebvas
Presumed	Austria	23.52500	2.4159037	23875.85	1875.357	149.86541	768.8462
Presumed	Belgium	21.90000	1.9357874	22499.62	1958.357	154.69504	593.8462
Presumed	Finland	18.44167	1.5264089	21018.92	1615.286	93.57447	771.3846
Presumed	France	16.75833	1.5974174	22602.85	2159.643	156.15327	432.6923
Presumed	Italy	11.10000	4.2769998	21554.15	1757.000	121.94294	712.1538
Presumed	Norway	15.44167	1.1090195	26448.38	2217.214	69.99821	661.6154
Presumed	Spain	28.10833	4.9630376	16933.00	1289.071	161.11430	654.7692
Presumed	Sweden	13.12500	1.7535030	22415.46	1951.357	72.34575	595.3077
Presumed	Switzerland	14.18250	1.7090940	27233.00	2776.071	96.38543	423.5385

```
p <- ggplot(data = by_country,
            mapping = aes(x = don.rate,
                          y = reorder(country, don.rate),
                          color = consent.law))
p +
  geom_point(size=3) +
  labs(x="Donor Procurement Rate",
       y="",
       color="Consent Law") +
  scale_color_manual(values = ubdc_palette[c(1,5)]) +
  theme(legend.position = "top") +
  theme(panel.grid.major.y = element_line(color = "grey80", size = 0.1)) #optional
```



Clevland using facet

```
p <- ggplot(data = by_country,
            mapping = aes(x = don.rate,
                          y = reorder(country, don.rate)))

p + geom_point(size = 3) +
  facet_wrap(~ consent.law, scales = "free_y", ncol = 1) +
  labs(x = "Donor Procurement Rate",
       y = "")
```

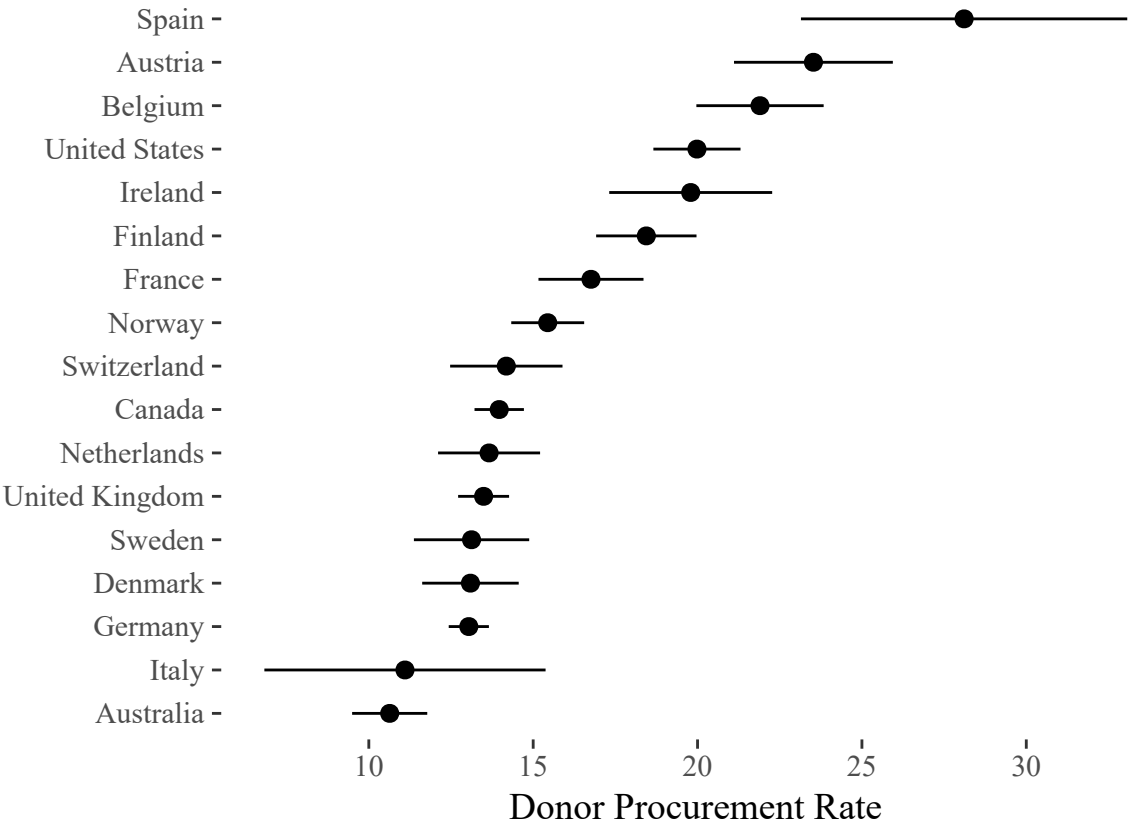


Clevland with error bars

Add bars with e.g. standard deviation

Use `geom_pointrange` and add `sd` measurement from summary table

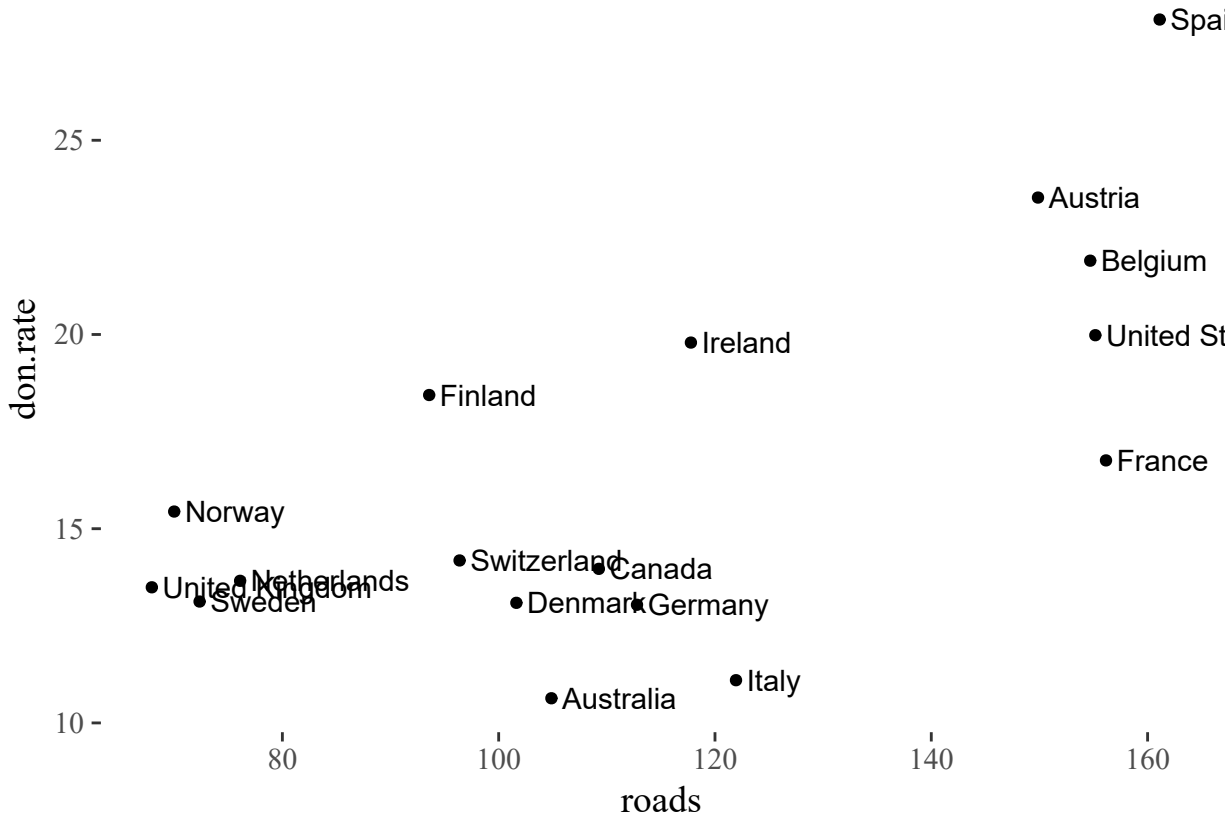
```
p <- ggplot(by_country, aes(reorder(country, don.rate), don.rate))

p + geom_pointrange(aes(ymin = don.rate - don.sd, ymax = don.rate + don.sd)) +
  labs(
    x = "",
    y = "Donor Procurement Rate"
  ) +
  coord_flip()
```



5.2 Plotting Text

Simple options

```
p <- ggplot(by_country, aes(roads, don.rate)) +
  geom_point() +
  geom_text(aes(x = roads + 1, label = country), hjust = 0)
#Roads + 1 moves data off the points
#hjust = 0 LEFT justifies
#hjust = 1 RIGHT justifies
p
```



Still messy - up the ante

Text with `ggrepel`

New data

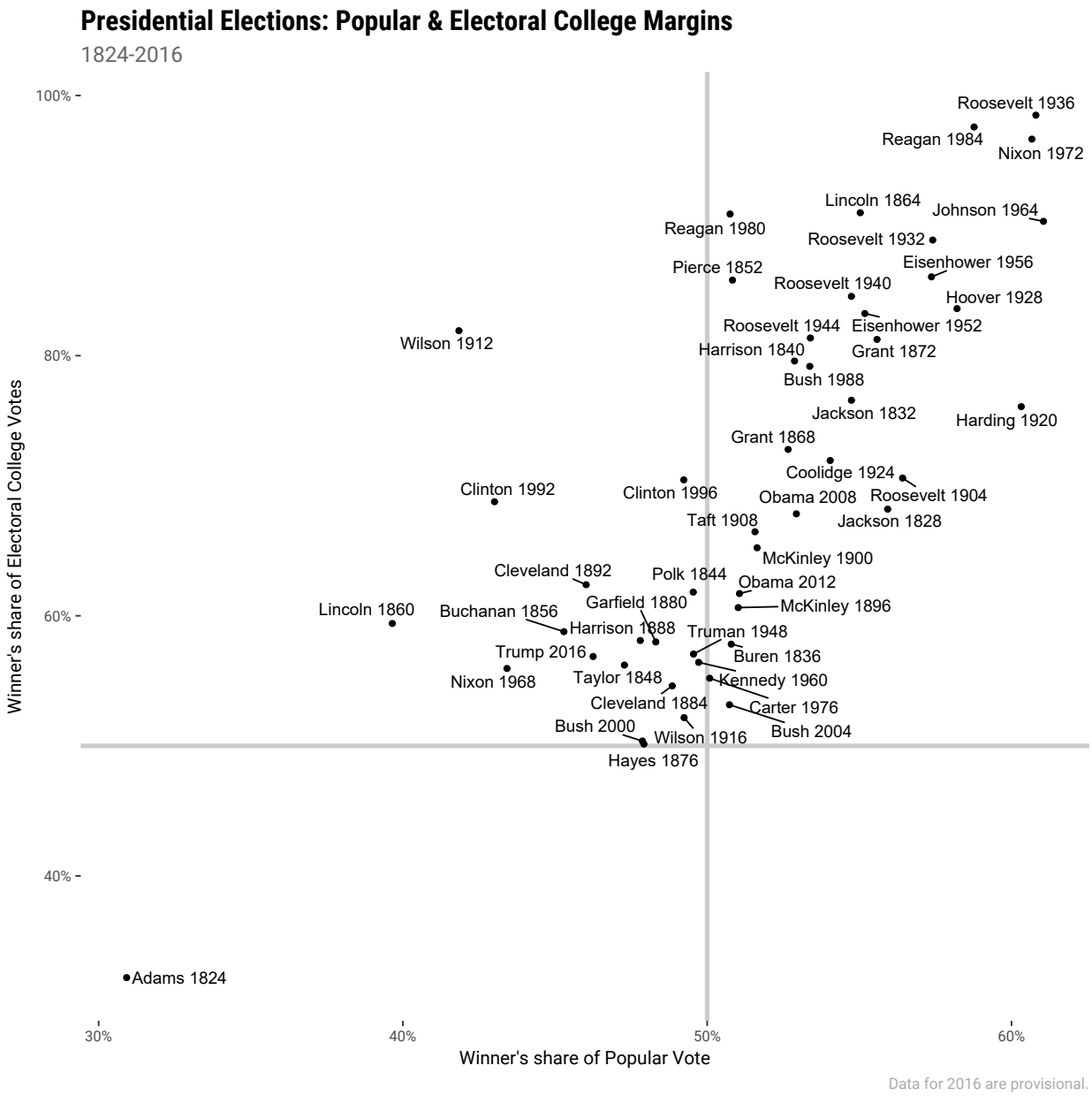
```
library(ggrepel)

elections_historic <- elections_historic
elections_historic %>%
  select(1:7) %>%
  head(., n = 15)
```

election	year	winner	win_party	ec_pct	popular_pct	popular_margin
10	1824	John Quincy Adams	D.-R.	0.3218	0.3092	-0.1044
11	1828	Andrew Jackson	Dem.	0.6820	0.5593	0.1225
12	1832	Andrew Jackson	Dem.	0.7657	0.5474	0.1781
13	1836	Martin Van Buren	Dem.	0.5782	0.5079	0.1420
14	1840	William Henry Harrison	Whig	0.7959	0.5287	0.0605
15	1844	James Polk	Dem.	0.6182	0.4954	0.0145
16	1848	Zachary Taylor	Whig	0.5621	0.4728	0.0479
17	1852	Franklin Pierce	Dem.	0.8581	0.5083	0.0695
18	1856	James Buchanan	Dem.	0.5878	0.4529	0.1220
19	1860	Abraham Lincoln	Rep.	0.5941	0.3965	0.1013
20	1864	Abraham Lincoln	Rep.	0.9099	0.5503	0.1008

election	year	winner	win_party	ec_pct	popular_pct	popular_margin
21	1868	Ulysses Grant	Rep.	0.7279	0.5266	0.0532
22	1872	Ulysses Grant	Rep.	0.8125	0.5558	0.1180
23	1876	Rutherford Hayes	Rep.	0.5014	0.4792	-0.0300
24	1880	James Garfield	Rep.	0.5799	0.4831	0.0009

A nice plot coming up

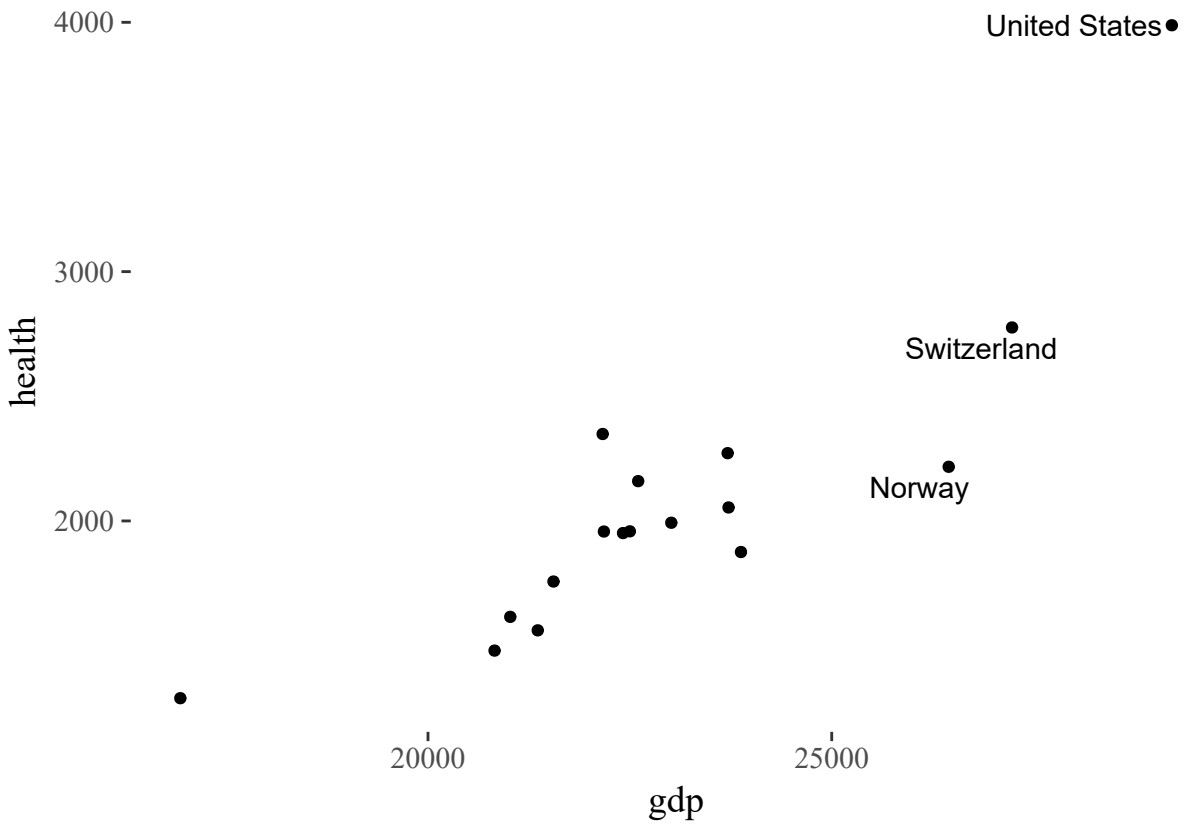
```
#pre-define labels
p_title <- "Presidential Elections: Popular & Electoral College Margins"
p_subtitle <- "1824-2016"
p_caption <- "Data for 2016 are provisional."
x_label <- "Winner's share of Popular Vote"
y_label <- "Winner's share of Electoral College Votes"

#Define the plot
p <- ggplot(elections_historic, aes(popular_pct, ec_pct, label = winner_label))

p +
  geom_hline(yintercept = 0.5, size = 1.4, color = "grey80") +  #50% x and y lines
  geom_vline(xintercept = 0.5, size = 1.4, color = "grey80") +
  geom_point() +
  geom_text_repel() +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::percent) +
  labs(
    x = x_label,
    y = y_label,
    title = p_title,
    subtitle = p_subtitle,
    caption = p_caption) +
  theme(text = element_text(size = 12, family = "Roboto"),
        plot.subtitle = element_text(color="#666666", size = 14),
        plot.title = element_text(family="Roboto Condensed", face = "bold", size = 18, hjust = 0),
        plot.caption = element_text(color="#AAAAAA", size=10))
```



5.3 Labelling outliers

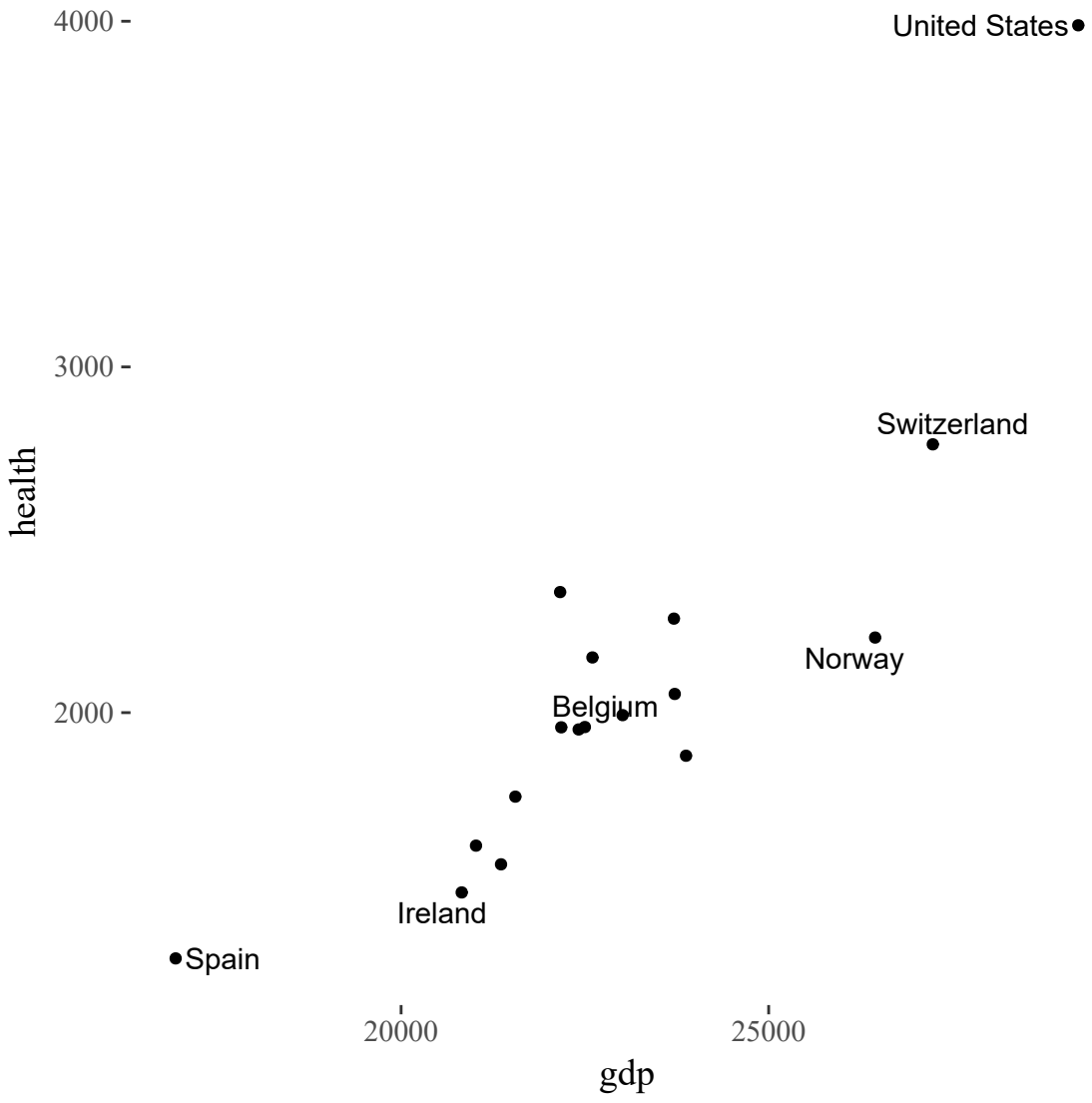
Use a different (filtered) dataframe for the labels

```
ggplot(by_country, aes(gdp, health)) +
  geom_point() +
  geom_text_repel(data = by_country %>% filter(gdp > 25000),
                  mapping = aes(label = country))
```



Label outliers on multiple conditions

```
ggplot(by_country, aes(gdp, health)) +
  geom_point() +
  geom_text_repel(data = by_country %>%
                    filter(gdp > 25000 | health < 1500 | country %in% "Belgium"),
                  mapping = aes(label = country))
```



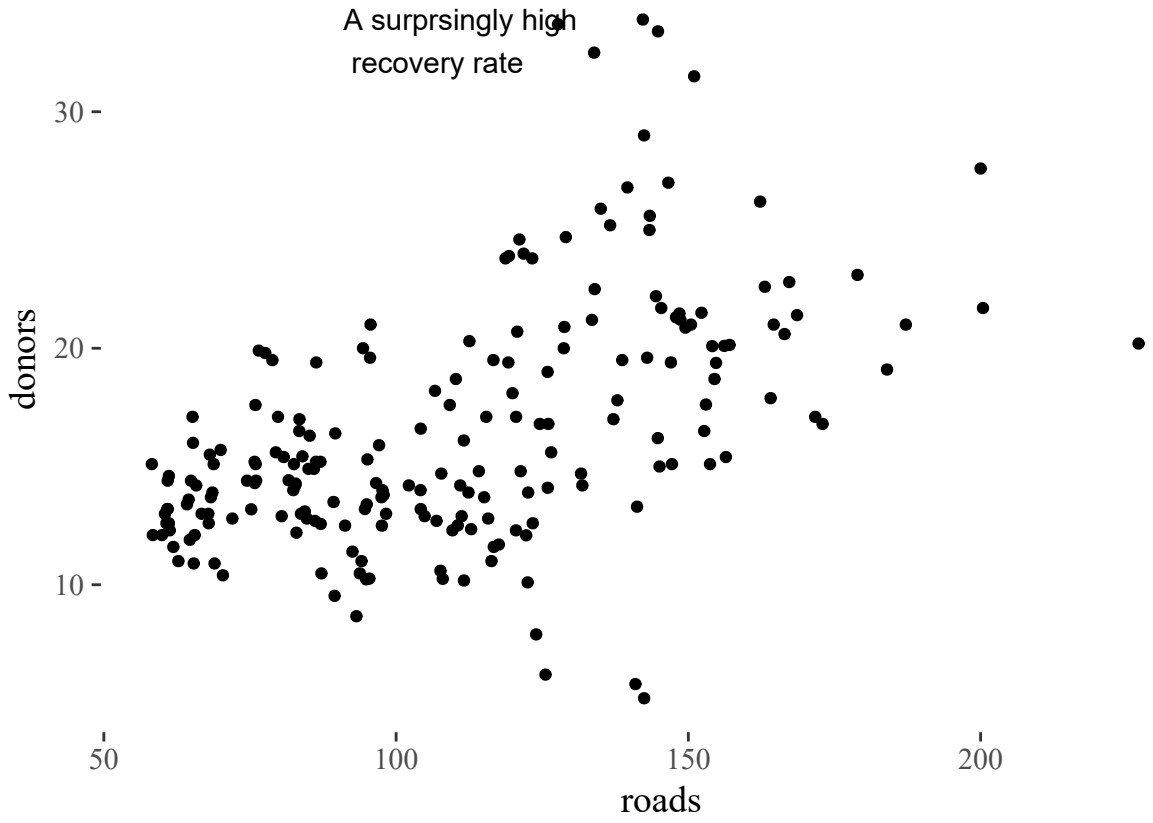
Label outliers using a dummy variable

```
#Create dummy variable
organdata$ind <- organdata$ccode %in% c("Ita", "Spa") & organdata$year > 1998

ggplot(organdata, aes(roads, donors, color = ind)) + #color by dummy variable
  geom_point() +
  geom_text_repel(data = organdata %>% filter(ind == TRUE),
                  mapping = aes(label = ccode)) +
  guides(color = FALSE) +
  scale_color_manual(values = ubdc_palette[c(1,5)])
```

5.4 Write and draw in the plot area

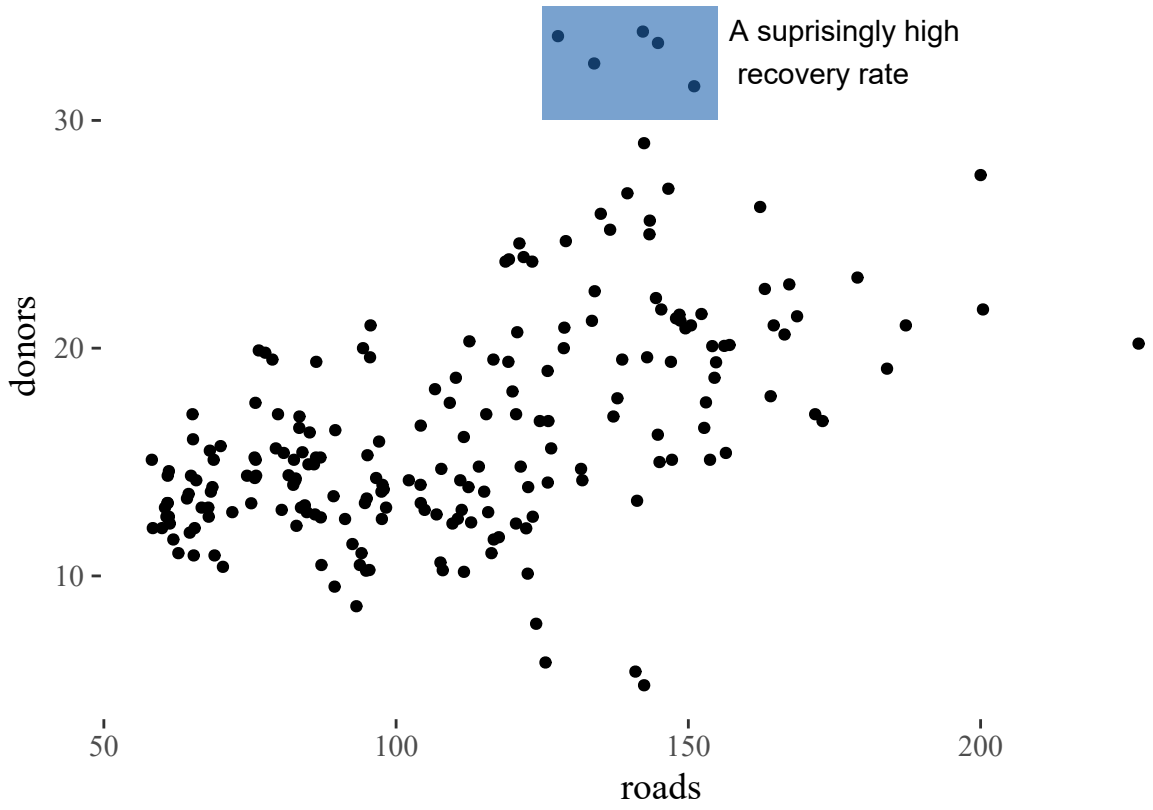
Annotate text

```
ggplot(organdata, aes(roads, donors)) +
  geom_point() +
  annotate(geom = "text", x = 91, y = 33,
           label = "A surprsingly high \n recovery rate",  #\n for newline
           hjust = 0)
```



Annotate a rectangle

```
ggplot(organdata, aes(roads, donors)) +
  geom_point() +
  annotate(geom = "rect", xmin = 125, xmax = 155,
           ymin = 30, ymax = 35, fill = ubdc_palette[6], alpha = 0.6) +
  annotate(geom = "text", label = "A suprisingly high \n recovery rate", x = 157, y = 33,
             fontfamily = "Roboto Condensed", hjust = 0)
```



5.5 Understand Scales, Guides and Themes