

## Wrangling and Graphing

```
data(anscombe)
```

```
ans <- anscombe
```

```
str(ans)
```

```
'data.frame':  11 obs. of  8 variables:
 $ x1: num  10 8 13 9 11 14 6 4 12 7 ...
 $ x2: num  10 8 13 9 11 14 6 4 12 7 ...
 $ x3: num  10 8 13 9 11 14 6 4 12 7 ...
 $ x4: num   8 8 8 8 8 8 8 19 8 8 ...
 $ y1: num  8.04 6.95 7.58 8.81 8.33 ...
 $ y2: num  9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...
 $ y3: num  7.46 6.77 12.74 7.11 7.81 ...
 $ y4: num  6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...
```

```
ans <- as_tibble(anscombe)
```

```
glimpse(ans)
```

```
Observations: 11
Variables: 8
 $ x1 <dbl> 10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5
 $ x2 <dbl> 10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5
 $ x3 <dbl> 10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5
 $ x4 <dbl> 8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8
 $ y1 <dbl> 8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68
 $ y2 <dbl> 9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74
 $ y3 <dbl> 7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73
 $ y4 <dbl> 6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89
```

```
ans %>%
```

```
  summarize(mean.x1 = mean(x1),
            mean.x2 = mean(x2),
            mean.y1 = mean(y1),
            mean.y2 = mean(y2))
```

```
# A tibble: 1 x 4
```

```
  mean.x1 mean.x2 mean.y1 mean.y2
    <dbl>   <dbl>   <dbl>   <dbl>
1       9       9     7.50     7.50
```

```
ans %>%
```

```
  summarize(sd.x1 = sd(x1),
            sd.x2 = sd(x2),
```

Table 1: Regressions of y1 on x1 and y2 on x2

	mod1	mod2
(Intercept)	3.00*	3.00*
	(1.12)	(1.12)
x1	0.50**	
	(0.12)	
x2		0.50**
		(0.12)
R <sup>2</sup>	0.67	0.67
Adj. R <sup>2</sup>	0.63	0.63
Num. obs.	11	11
RMSE	1.24	1.24

\*\*\* $p < 0.001$ , \*\* $p < 0.01$ , \* $p < 0.05$ 

```
sd.y1 = sd(y1),
sd.y2 = sd(y2))
```

```
# A tibble: 1 x 4
```

```
sd.x1 sd.x2 sd.y1 sd.y2
<dbl> <dbl> <dbl> <dbl>
```

```
1 3.32 3.32 2.03 2.03
```

```
mod1 <- lm(y1 ~ x1, data = ans)
```

```
mod2 <- lm(y1 ~ x2, data = ans)
```

```
p1 <- ggplot(ans, aes(x = x1, y = y1)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

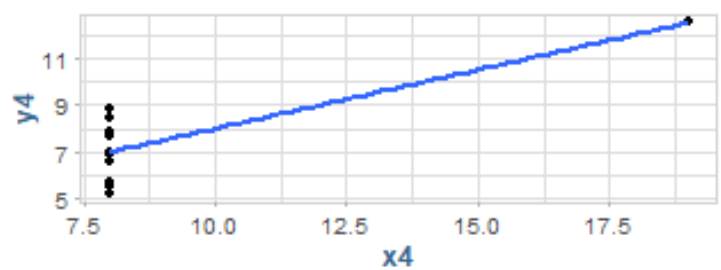
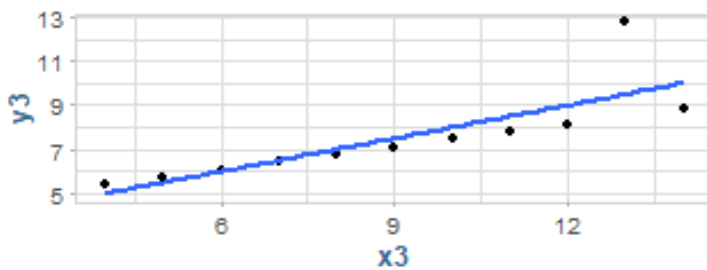
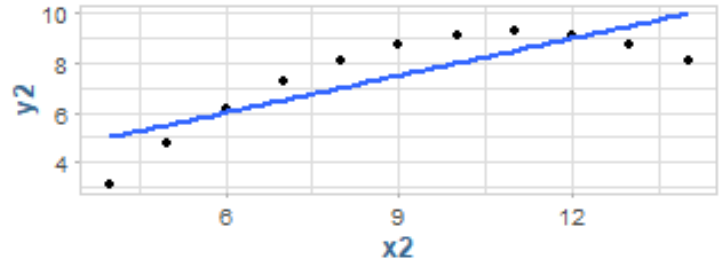
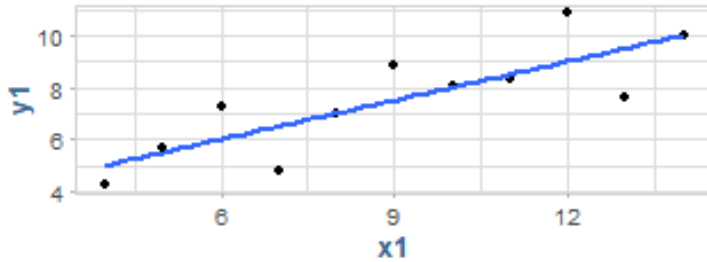
```
p2 <- ggplot(ans, aes(x = x2, y = y2)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

```
p3 <- ggplot(ans, aes(x = x3, y = y3)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

```
p4 <- ggplot(ans, aes(x = x4, y = y4)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

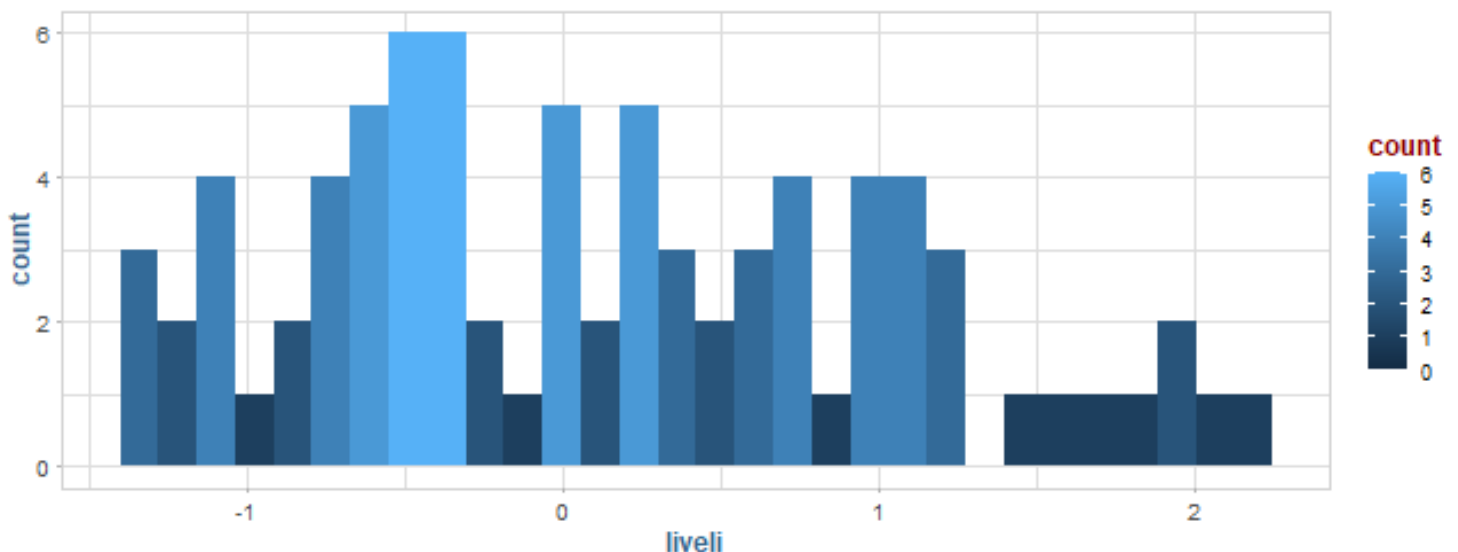
```
gridExtra::grid.arrange(p1, p2, p3, p4, ncol = 2)
```

```
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
```

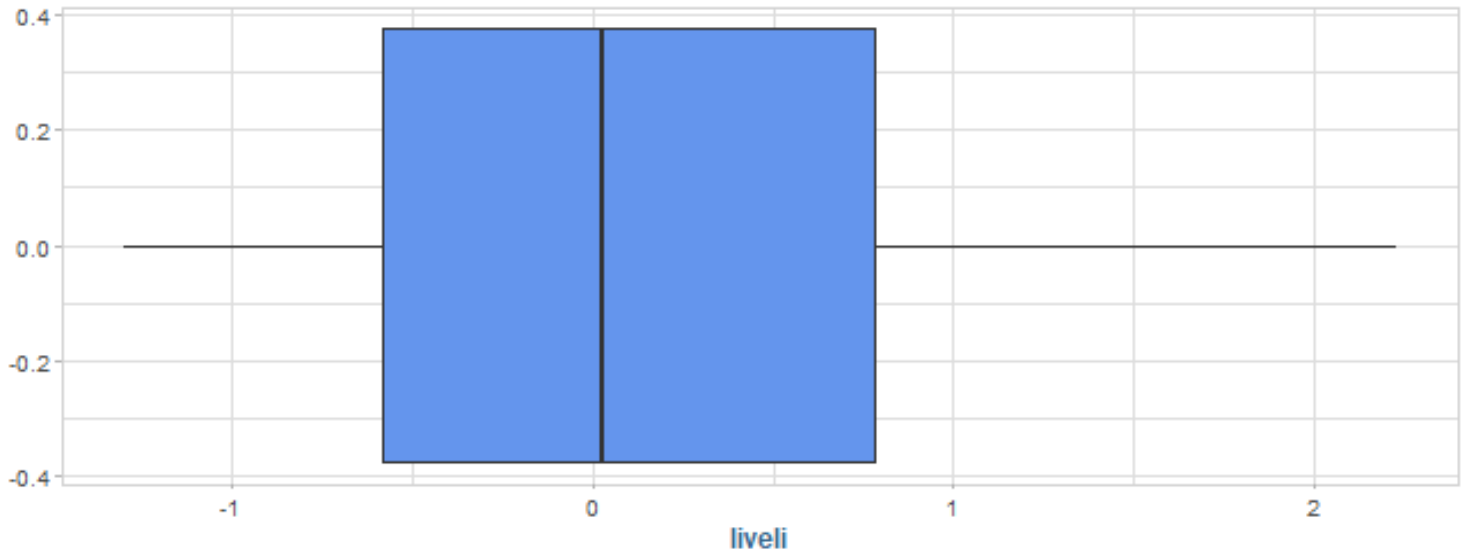


```
ifri <- get_csv("ifri_car_liv.csv")
ifri <- ifri[1:80,]
ifri <- ifri %>%
  rename(carbon = zbio, liveli = zliv)

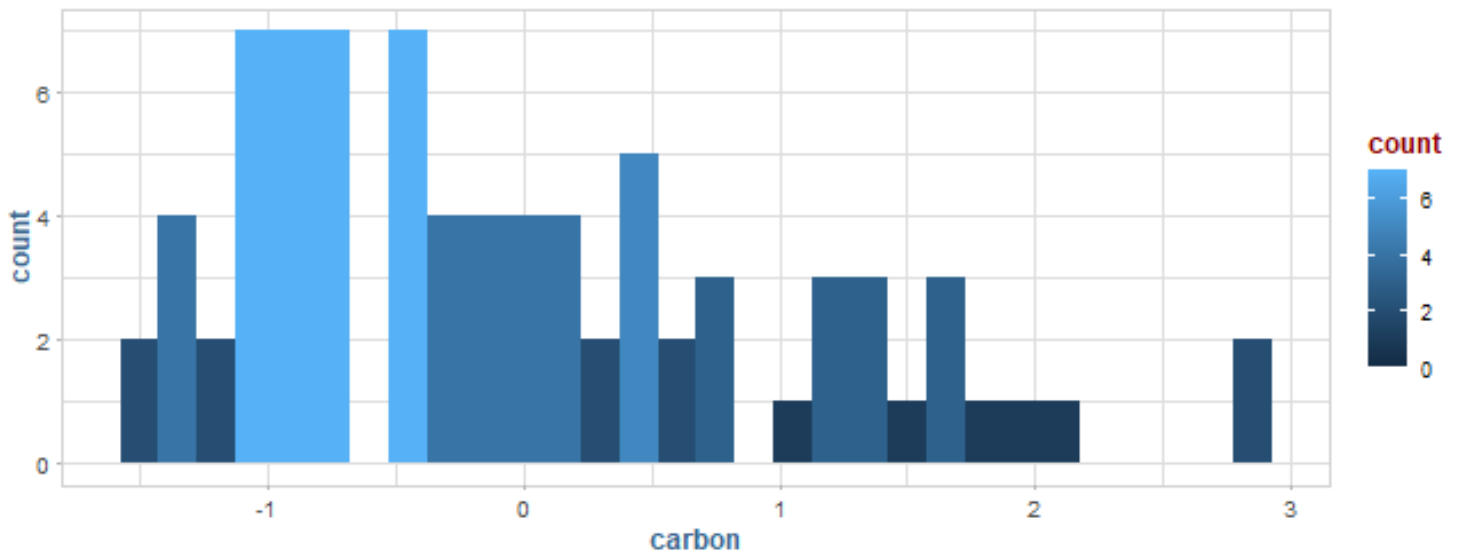
ggplot(ifri, aes(x = liveli)) +
  geom_histogram(aes(fill = ..count..), bins = 30)
```



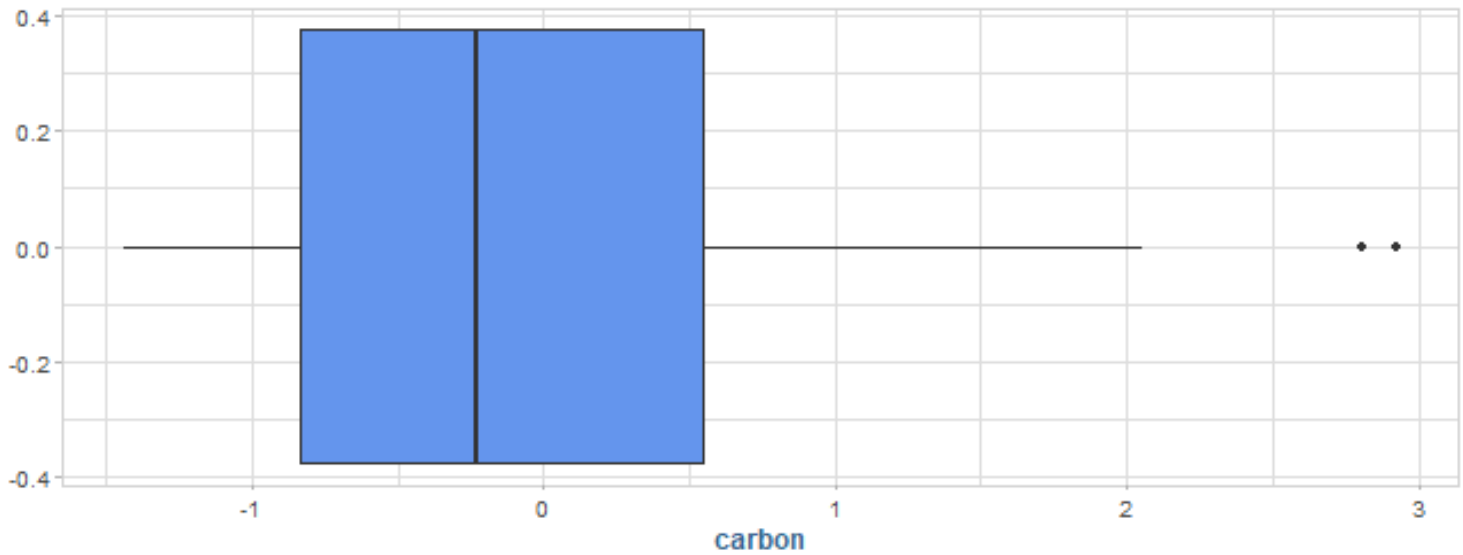
```
ggplot(efri, aes(y = liveli)) +  
  geom_boxplot(fill = "cornflowerblue") +  
  coord_flip()
```



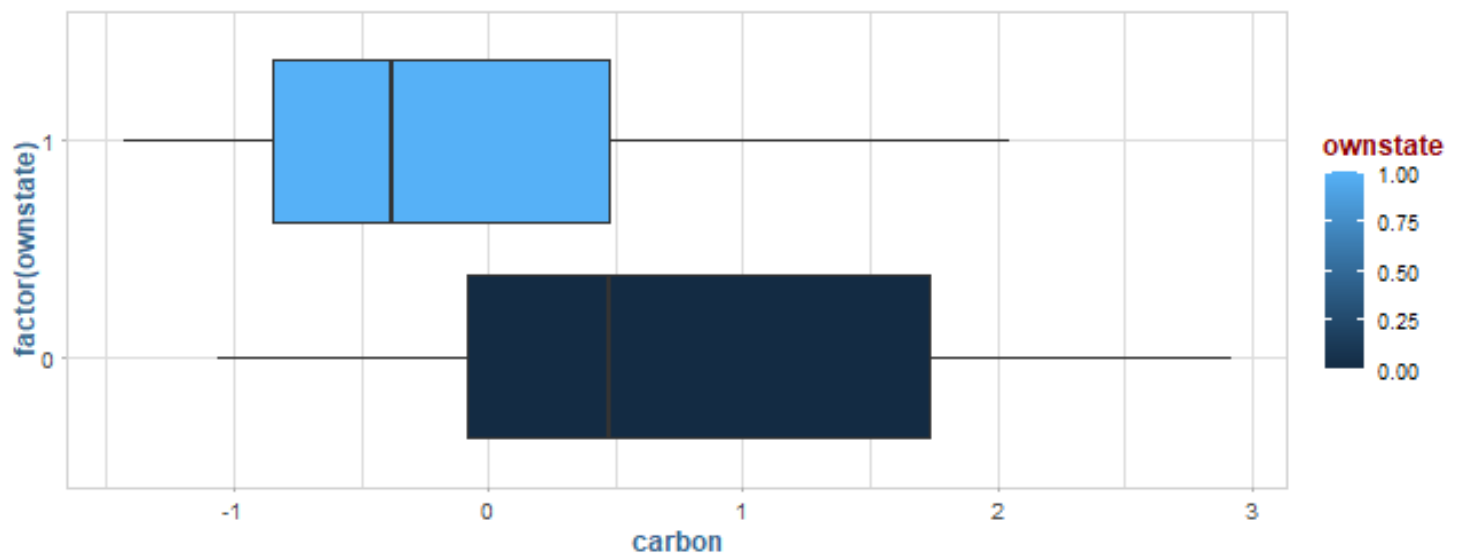
```
ggplot(efri, aes(x = carbon)) +  
  geom_histogram(aes(fill = ..count..), bins = 30)
```



```
ggplot(efri, aes(y = carbon)) +  
  geom_boxplot(fill = "cornflowerblue") +  
  coord_flip()
```



```
ggplot(ifri, aes(x= factor(ownstate),
                 y = carbon)) +
  geom_boxplot(aes(fill = ownstate)) +
  coord_flip()
```

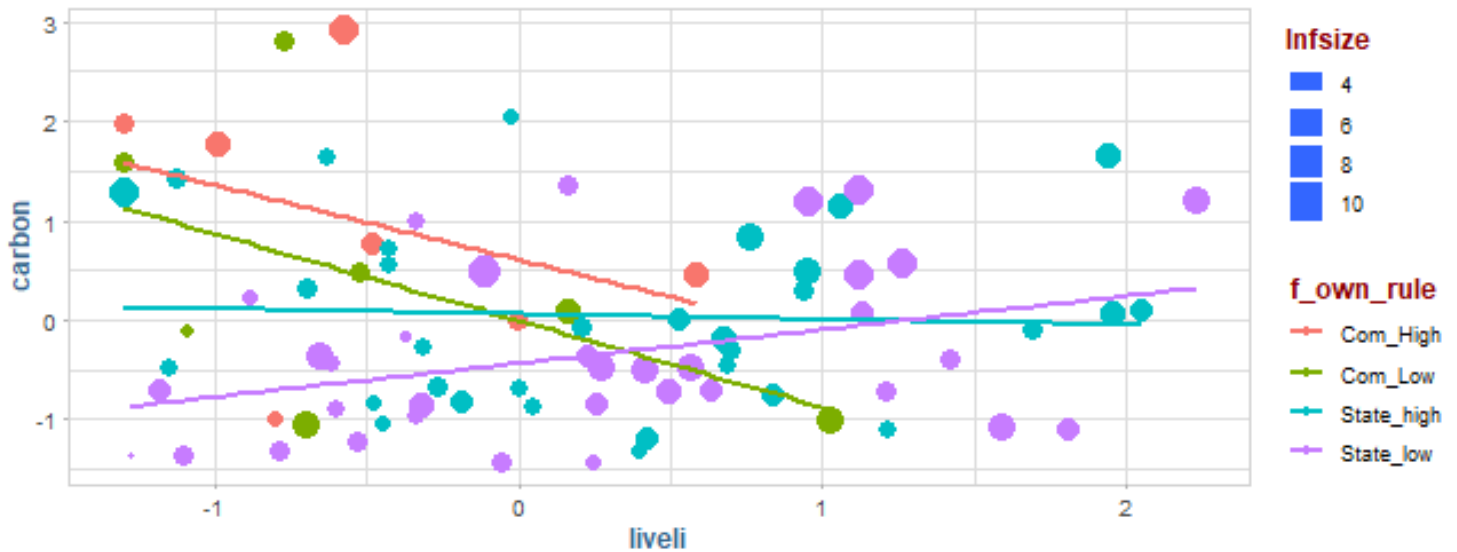


```
ifri2 <- mutate(ifri, f_own_rule =
  ifelse(ownstate == 1 & rulematch == 0, "State_low",
  ifelse(ownstate == 1 & rulematch == 1, "State_high",
  ifelse(ownstate == 0 & rulematch == 1, "Com_High",
  "Com_Low" ))))

ggplot(ifri2, aes(x = liveli,
                  y = carbon,
                  size = lnfsiz,
```

```
colour = f_own_rule)) +
geom_point() +
geom_smooth(method = "lm", se = F)
```

`geom\_smooth()` using formula 'y ~ x'



```
WDIsearch("gdp.*capita.*PPP")
```

```
indicator
[1,] "6.0.GDPpc_constant"
[2,] "NY.GDP.PCAP.PP.KD.ZG"
[3,] "NY.GDP.PCAP.PP.KD.87"
[4,] "NY.GDP.PCAP.PP.KD"
[5,] "NY.GDP.PCAP.PP.CD"
name
[1,] "GDP per capita, PPP (constant 2011 international $) "
[2,] "GDP per capita, PPP annual growth (%)"
[3,] "GDP per capita, PPP (constant 1987 international $)"
[4,] "GDP per capita, PPP (constant 2011 international $)"
[5,] "GDP per capita, PPP (current international $)"
```

```
WDIsearch("Co2.*capita")
```

```
indicator
[1,] "EN.ATM.CO2E.PC"
[2,] "EN.ATM.NOXE.PC"
[3,] "EN.ATM.METH.PC"
name
[1,] "CO2 emissions (metric tons per capita)"
[2,] "Nitrous oxide emissions (metric tons of CO2 equivalent per capita)"
```

```
[3,] "Methane emissions (kt of CO2 equivalent per capita)"
```

```
wdi_data <- WDI(indicator =
  c("NY.GDP.PCAP.PP.KD",
    "EN.ATM.CO2E.PC"),
  start = 2010,
  end = 2010,
  extra = T)
```

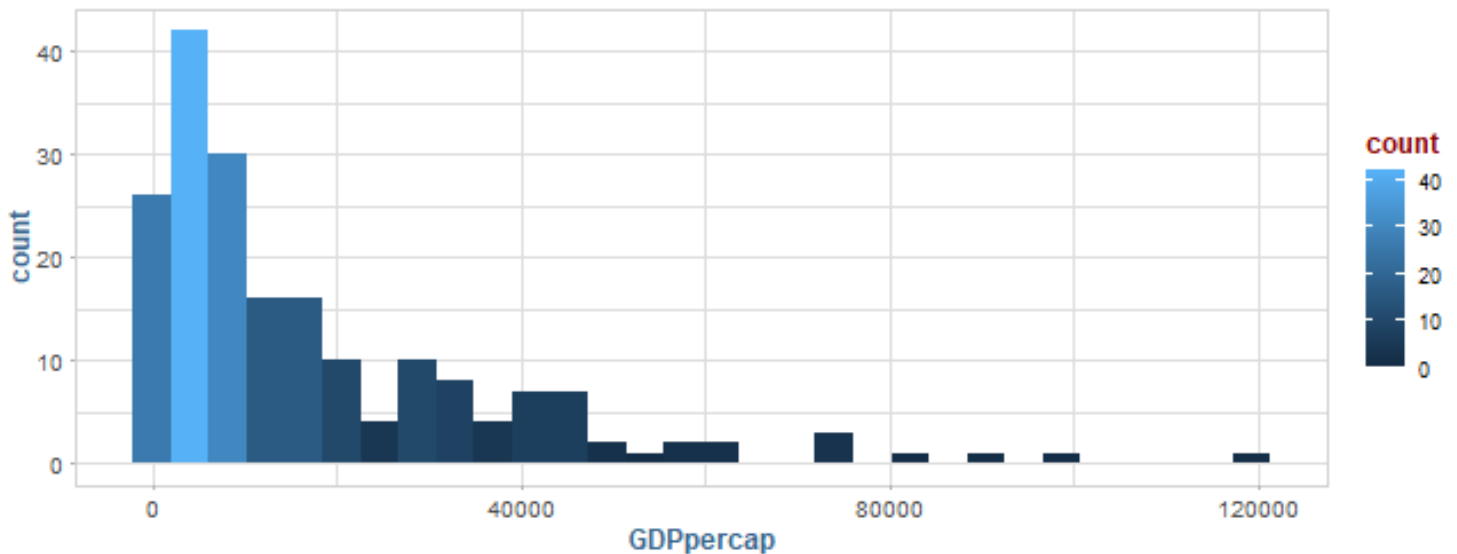
```
wdi_data <- wdi_data %>%
  filter(region != "Aggregates")
```

```
wdi <- wdi_data %>%
  rename(GDPpercap =
    NY.GDP.PCAP.PP.KD,
    Emit_CO2percap =
    EN.ATM.CO2E.PC)
```

```
ggplot(wdi, aes(x = GDPpercap)) +
  geom_histogram(aes(fill = ..count..))
```

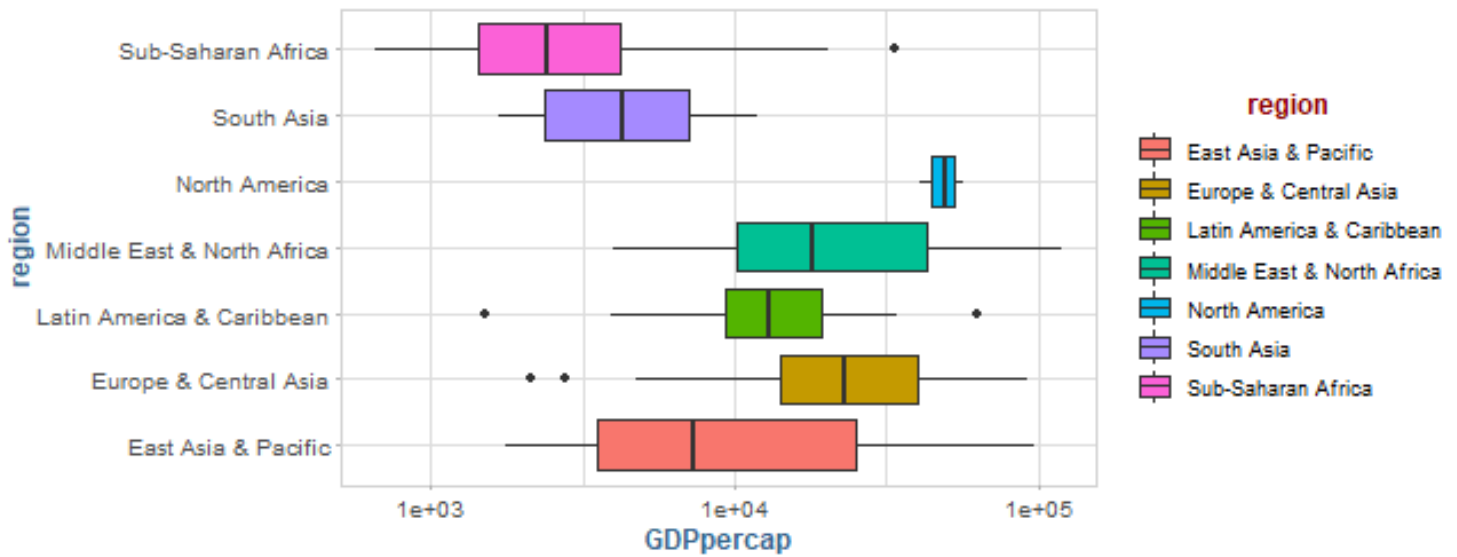
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 21 rows containing non-finite values (stat\_bin).



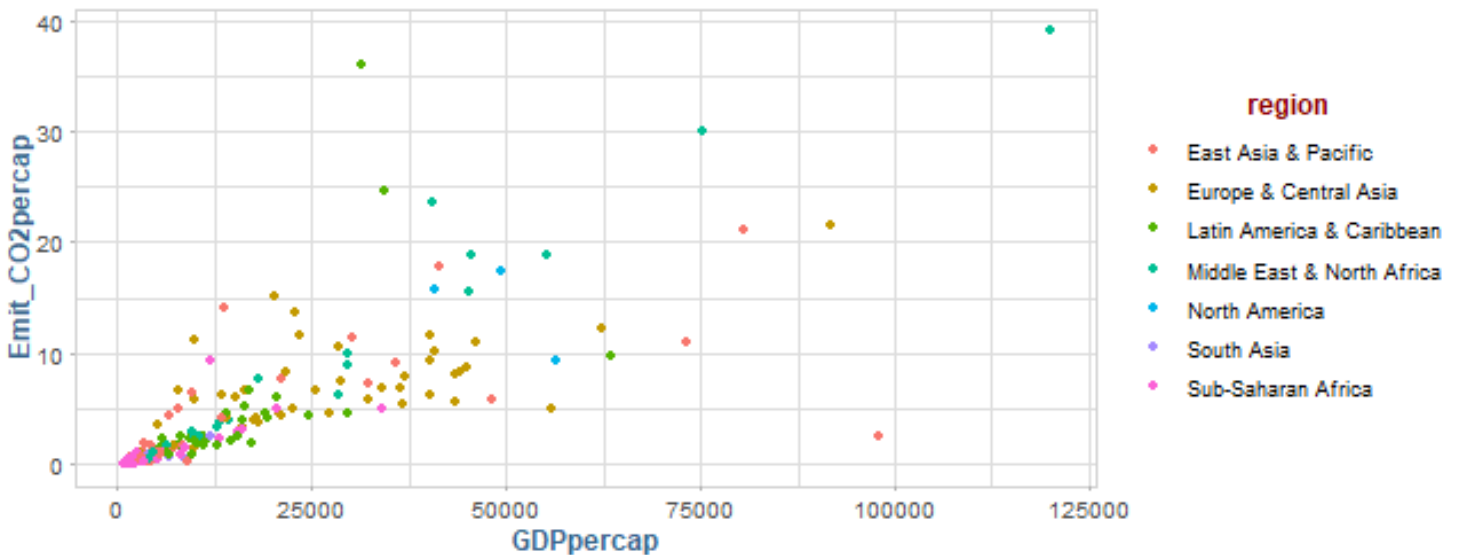
```
ggplot(wdi, aes(y = GDPpercap,
  x = region)) +
  geom_boxplot(aes(fill = region)) +
  coord_flip() +
  scale_y_log10()
```

Warning: Removed 21 rows containing non-finite values (stat\_boxplot).



```
ggplot(wdi, aes(x = GDPpercap, y = Emit_CO2percap)) +  
  geom_point(aes(color = region))
```

Warning: Removed 27 rows containing missing values (geom\_point).



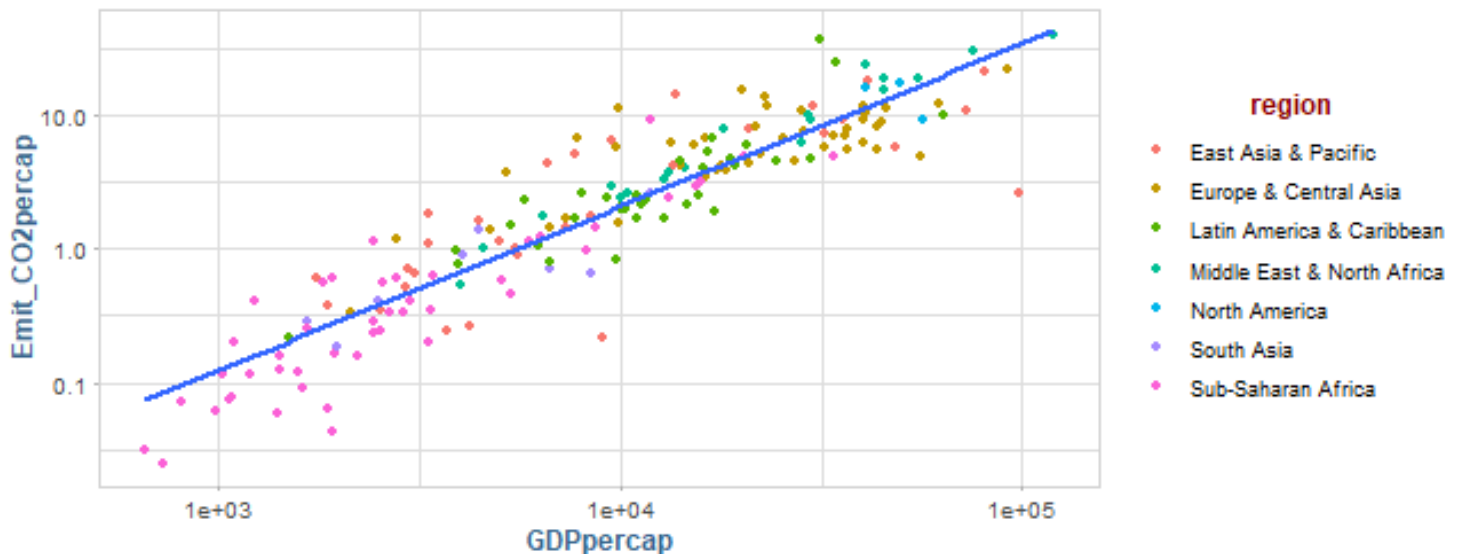
```
ggplot(wdi, aes(x = GDPpercap, y = Emit_CO2percap)) +  
  geom_point(aes(color = region)) +  
  scale_x_log10() +  
  scale_y_log10() +  
  geom_smooth(method = "lm", se = F)
```

`geom\_smooth()` using formula 'y ~ x'



Warning: Removed 27 rows containing non-finite values (stat\_smooth).

Warning: Removed 27 rows containing missing values (geom\_point).



```
dat_map <- map_data("world")
```

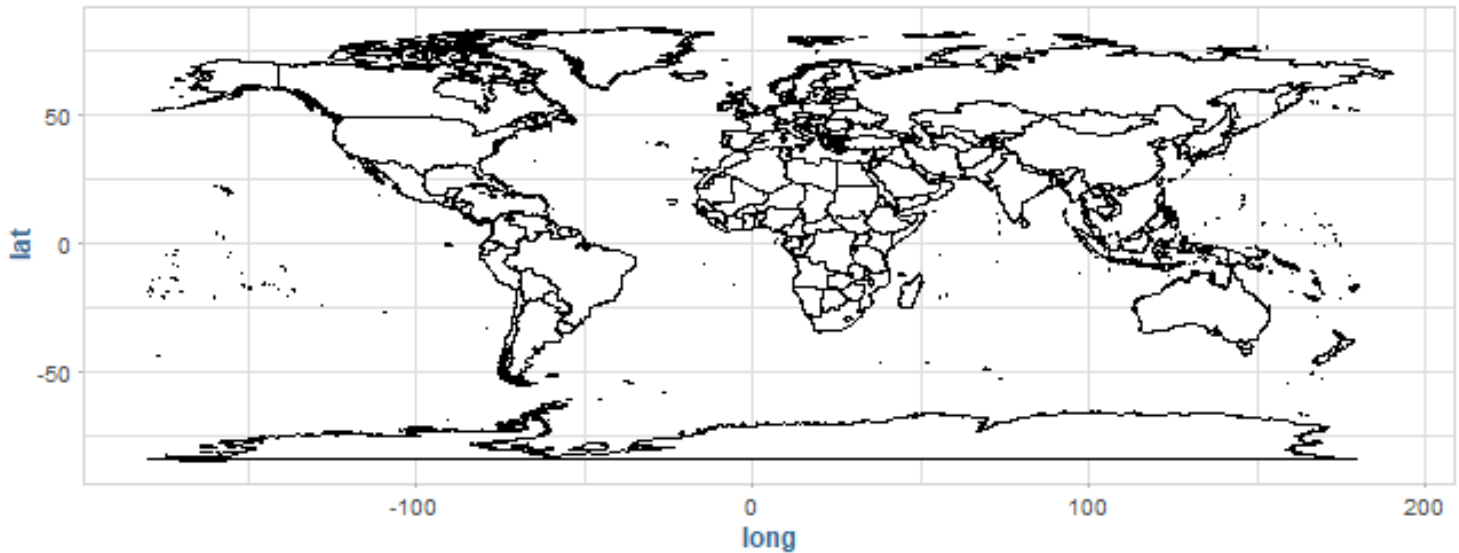
```
dim(dat_map)
```

```
[1] 99338      6
```

```
head(dat_map)
```

	long	lat	group	order	region	subregion
1	-69.89912	12.45200	1	1	Aruba	<NA>
2	-69.89571	12.42300	1	2	Aruba	<NA>
3	-69.94219	12.43853	1	3	Aruba	<NA>
4	-70.00415	12.50049	1	4	Aruba	<NA>
5	-70.06612	12.54697	1	5	Aruba	<NA>
6	-70.05088	12.59707	1	6	Aruba	<NA>

```
ggplot(dat_map, aes(x = long, y = lat, group = group)) +  
  geom_polygon(fill = "white", color = "black")
```



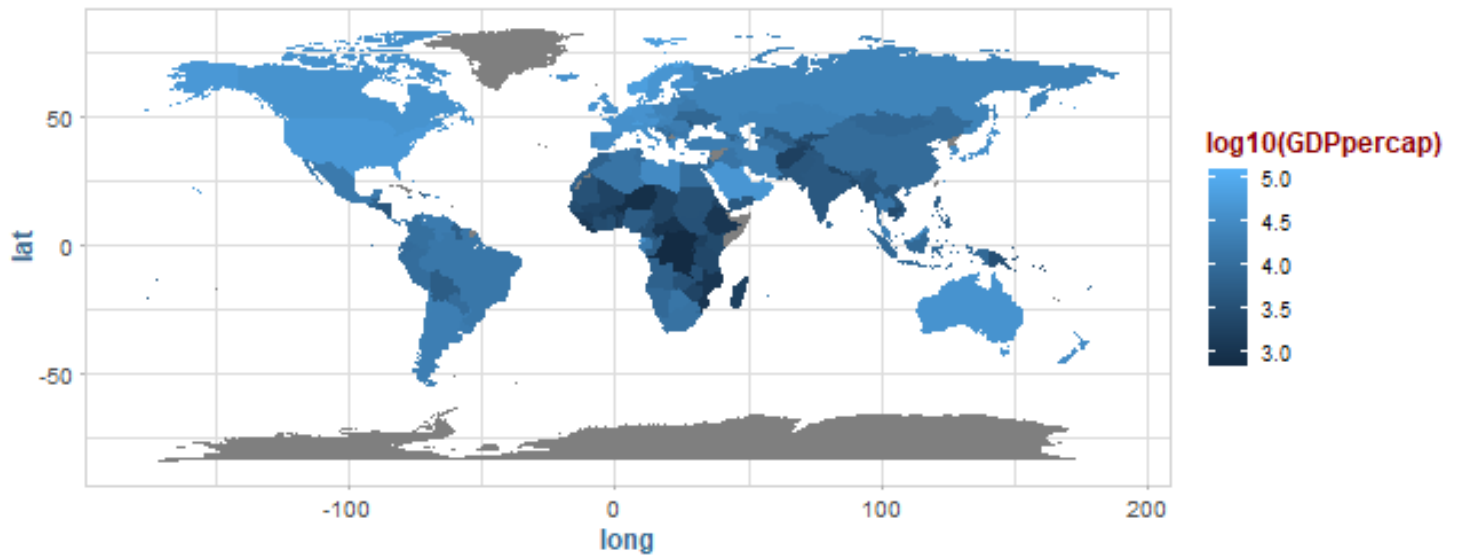
```
dat_map$ccode <- countrycode(dat_map$region,  
                             origin = "country.name",  
                             destination = "wb")
```

Warning in countrycode(dat\_map\$region, origin = "country.name", destination = "wb"): Some values were not found

```
wdi$ccode <- countrycode(wdi$country,  
                        origin = "country.name",  
                        destination = "wb")
```

```
merged <- full_join(dat_map, wdi, by = "ccode")
```

```
ggplot(merged, aes(x = long, y = lat,  
                  group = group, fill = log10(GDPpercap))) +  
  geom_polygon()
```



```
ggplot(merged, aes(x = long, y = lat,  
  group = group,  
  fill = log10(Emit_CO2percap))) +  
  geom_polygon() +  
  scale_fill_gradient(low = "green",  
    high = "red")
```

