Introduction to the data

COMMUNICATING WITH DATA IN THE TIDYVERSE

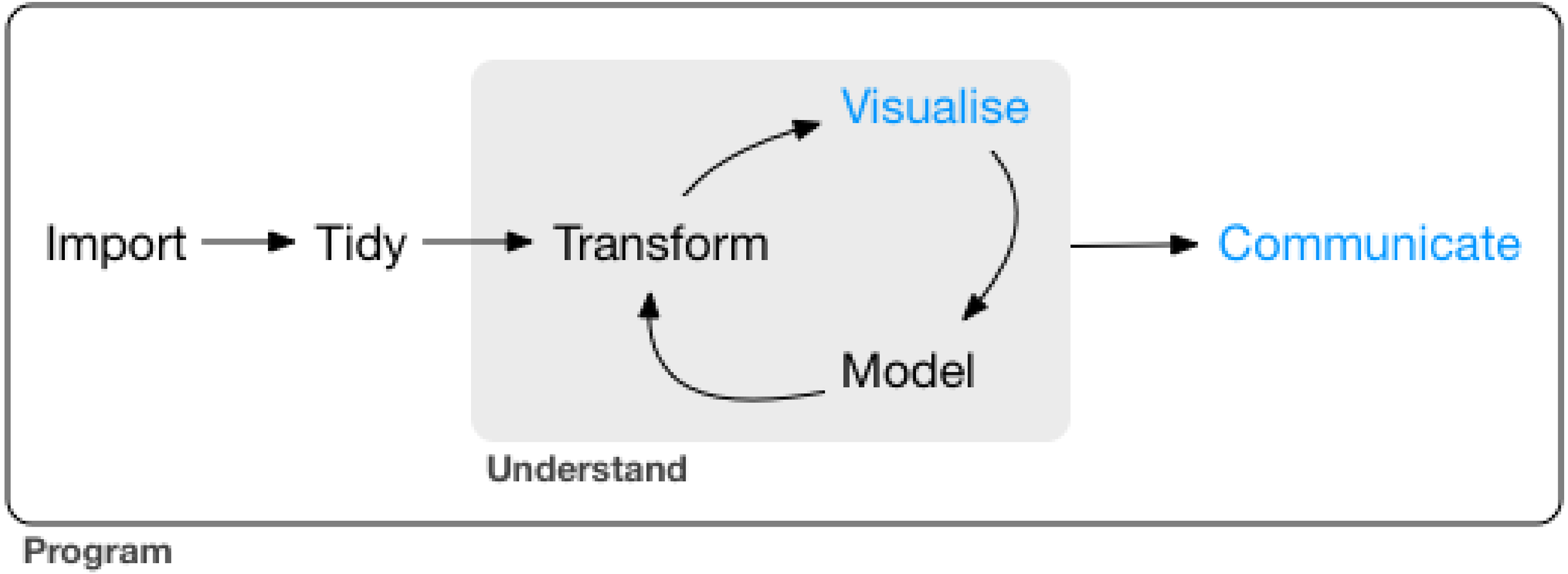
Timo Grossenbacher
Data Journalist

datacamp

This is me

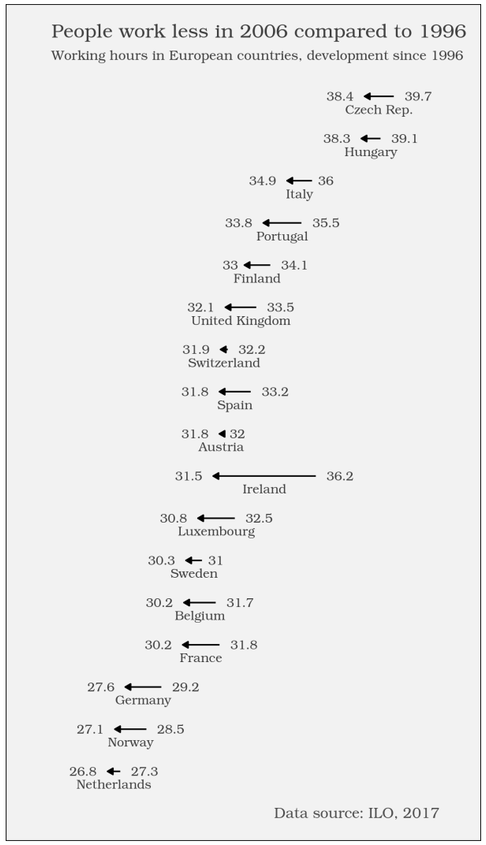
- Find examples of data journalism on **https://srfdata.github.io**

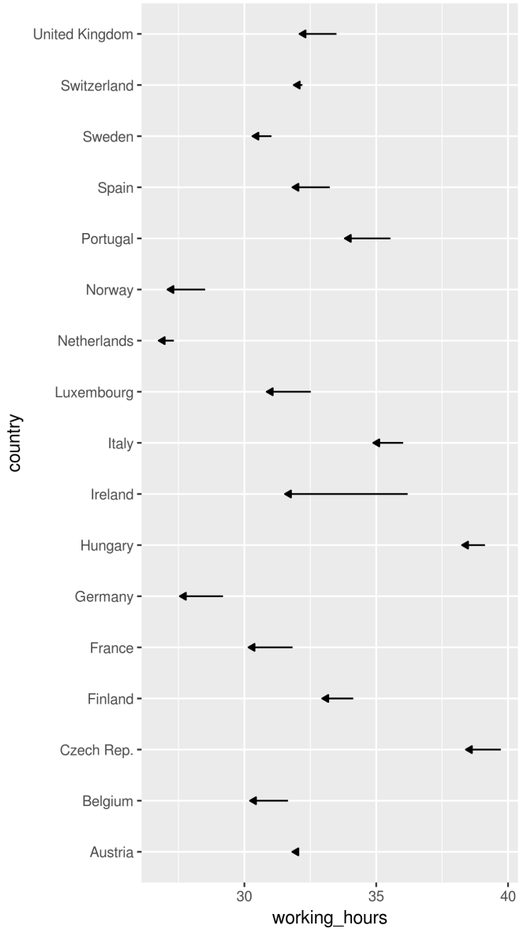
The last step in the Tidyverse process



[1] R for Data Science (http://r4ds.had.co.nz/communicate-intro.html)

What you are going to create





The reduction in weekly working hours in Europe

Code

Looking at the development between 1996 and 2006

Timo Grossenbacher

The data you are going to work with

```
ilo_working_hours
```

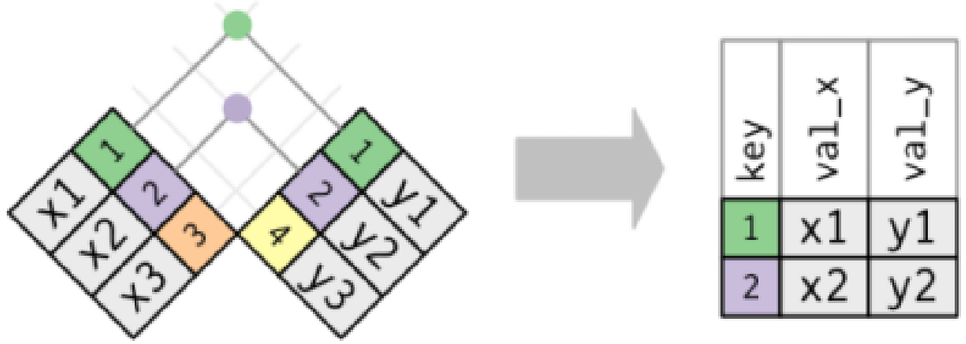
```
# A tibble: 737 x 3
       country   year working_hours
         <chr>  <chr>         <dbl>
 1   Australia 1980.0      34.57885
 2      Canada 1980.0      34.85000
 3     Denmark 1980.0      31.89808
 4     Finland 1980.0      35.56346
 5      France 1980.0      35.42308
 6     Iceland 1980.0      35.84615
 7       Italy 1980.0      35.74635
 8       Japan 1980.0      40.78846
 9 Korea, Rep. 1980.0      55.30769
10      Norway 1980.0      30.37885
# ... with 727 more rows
```

The data you are going to work with

```
ilo_hourly_compensation
```

```
# A tibble: 831 x 3
            country   year hourly_compensation
              <chr>  <chr>                <dbl>
 1        Australia 1980.0                 8.44
 2          Austria 1980.0                 8.87
 3          Belgium 1980.0                11.74
 4           Canada 1980.0                 8.87
 5          Denmark 1980.0                10.83
 6          Finland 1980.0                 8.61
 7           France 1980.0                 8.90
 8           Greece 1980.0                 3.72
 9 Hong Kong, China 1980.0                 1.50
10          Ireland 1980.0                 6.44
# ... with 821 more rows
```

The inner_join() verb / function



key	val_x	val_y
1	x1	y1
2	x2	y2

```
x %>%
  inner_join(y, by = "key")
```

```
#> # A tibble: 2 × 3
#>     key val_x val_y
#>   <dbl> <chr> <chr>
#> 1     1    x1    y1
#> 2     2    x2    y2
```

[1] R for Data Science (http://r4ds.had.co.nz/relational-data.html#inner-join)

Let's do this!

COMMUNICATING WITH DATA IN THE TIDYVERSE

Filtering and plotting the data

COMMUNICATING WITH DATA IN THE TIDYVERSE

Timo Grossenbacher
Data Journalist

Filter the data for European countries

```
ilo_data %>%
  filter(country == "Switzerland")
```

```
# A tibble: 27 x 4
       country   year hourly_compensation working_hours
        <fctr> <fctr>               <dbl>         <dbl>
 1 Switzerland   1980               10.96      34.70385
 2 Switzerland   1981               10.01      34.33462
 3 Switzerland   1982               10.31      34.12308
 4 Switzerland   1983               10.33      33.84231
 5 Switzerland   1984                9.52      33.47885
 6 Switzerland   1985                9.55      33.35961
 7 Switzerland   1986               13.62      33.19615
 8 Switzerland   1987               16.90      33.17308
 9 Switzerland   1988               17.81      33.16269
10 Switzerland   1989               16.54      32.87308
# ... with 17 more rows
```

```
ilo_data %>%
  filter(country %in% c("Sweden", "Switzerland"))
```

```
# A tibble: 54 x 4
      country   year hourly_compensation working_hours
       <fctr> <fctr>                <dbl>         <dbl>
1      Sweden   1980                12.40      29.16923
2 Switzerland   1980                10.96      34.70385
3      Sweden   1981                11.70      29.00769
4 Switzerland   1981                10.01      34.33462
5      Sweden   1982                 9.99      29.27885
# ... with 49 more rows
```

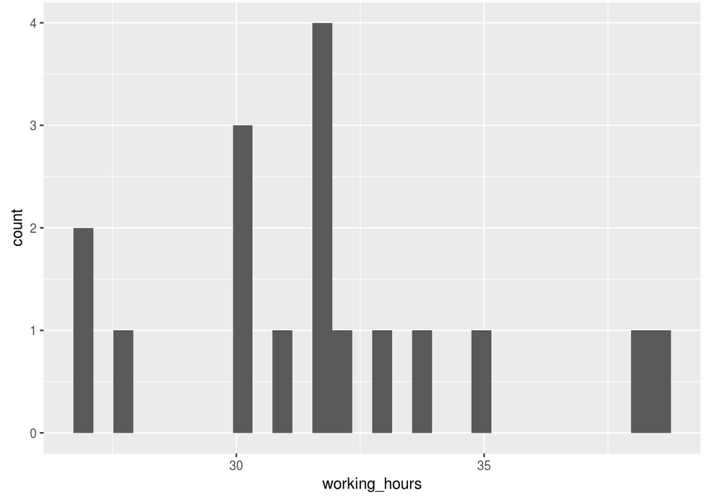
...equivalent to:

```
ilo_data %>%
  filter(country == "Sweden" | country == "Switzerland")
```

The relationship between both indicators

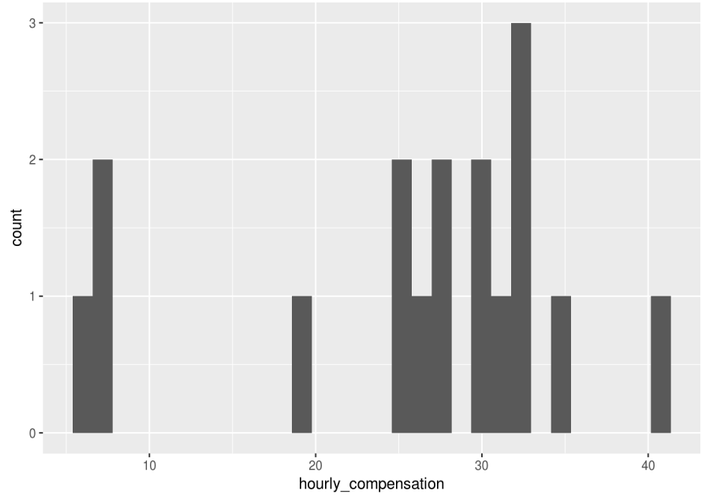
```
plot_data <-
  ilo_data %>%
    filter(year == 2006)

ggplot(plot_data) +
  geom_histogram(
    aes(x = working_hours))
```

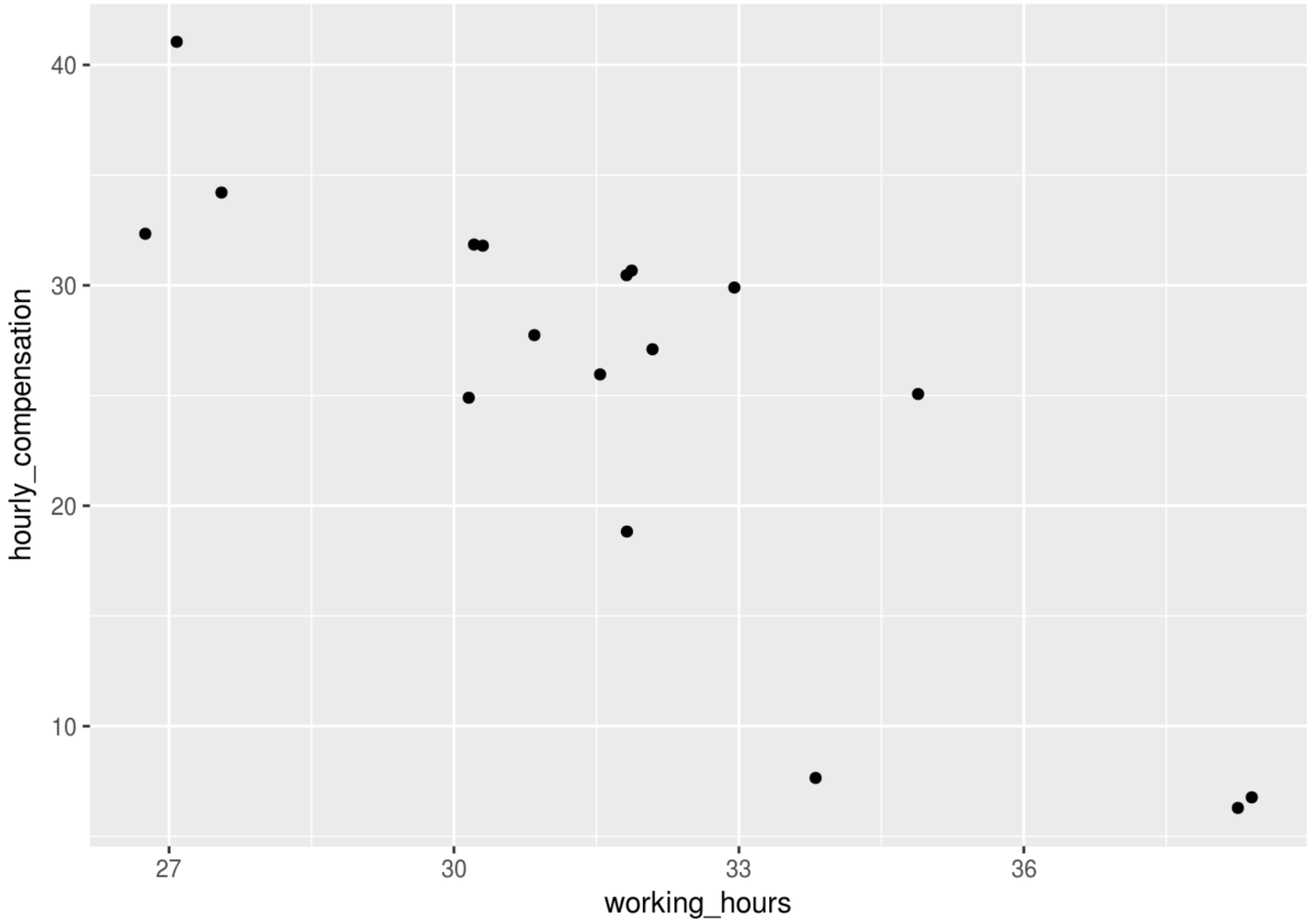



```
plot_data <-
  ilo_data %>%
    filter(year == 2006)

ggplot(plot_data) +
  geom_histogram(
    aes(x = hourly_compensation))
```



The relationship between both indicators



Adding labels to the plot

Some dplyr function repetition

```
ilo_data %>%
  group_by(country) %>%
  summarize(median_working_hours = median(working_hours))
```

```
# A tibble: 17 x 2
     country median_working_hours
      <fctr>                <dbl>
1    Austria             31.69904
2    Belgium             32.03846
3 Czech Rep.             39.10000
4    Finland             34.04808
5     France             32.34615
# ... with 12 more rows
```

Let's practice!

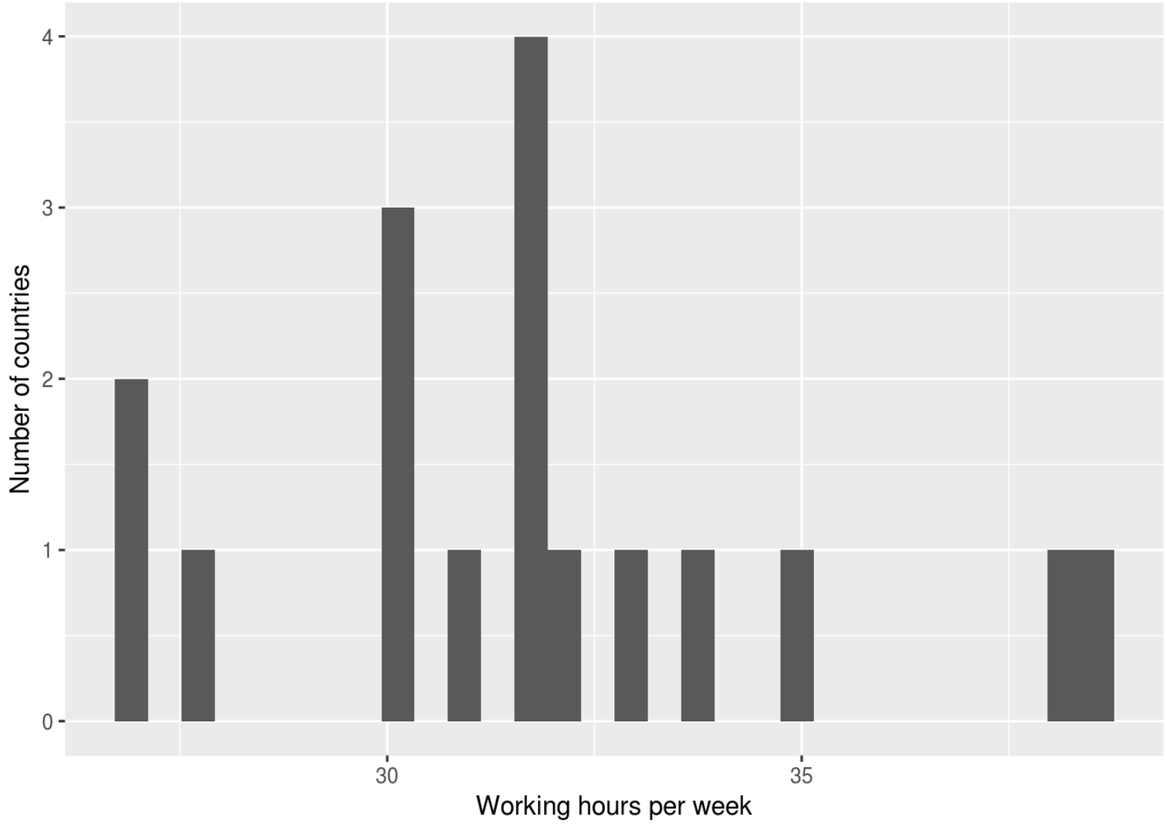
COMMUNICATING WITH DATA IN THE TIDYVERSE

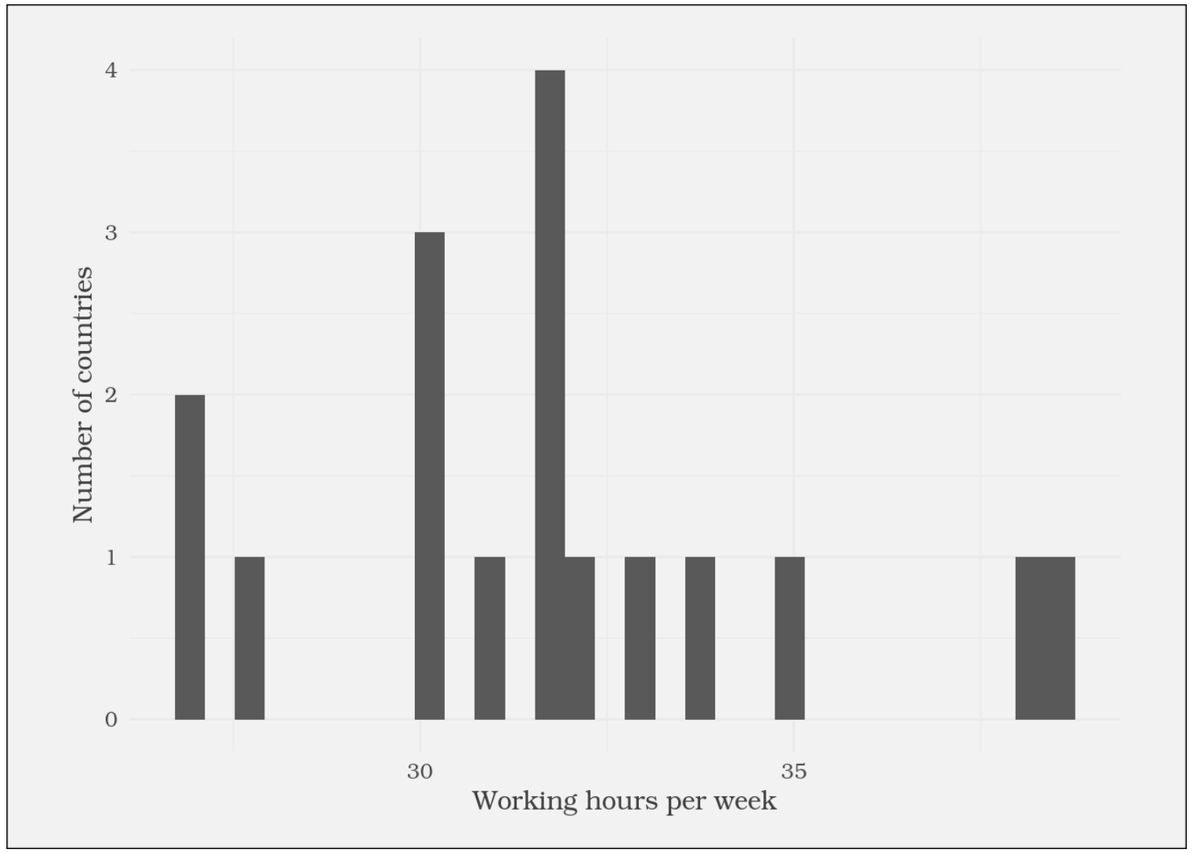
Custom ggplot2 themes

COMMUNICATING WITH DATA IN THE TIDYVERSE

Timo Grossenbacher
Data Journalist

The advantages of a custom look

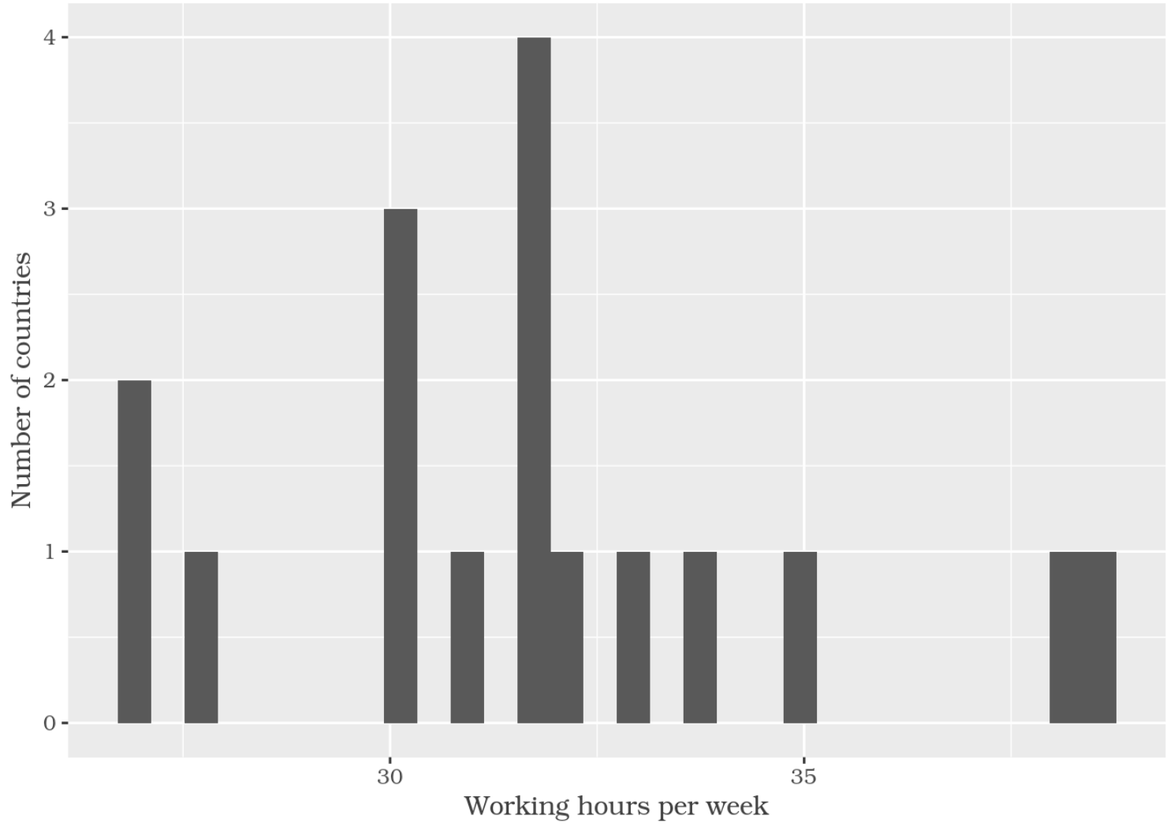




The theme() function

```
ggplot(plot_data) +
  geom_histogram(aes(
    x = working_hours)) +
  labs(x = "Working hours per week",
       y = "Number of countries") +
```

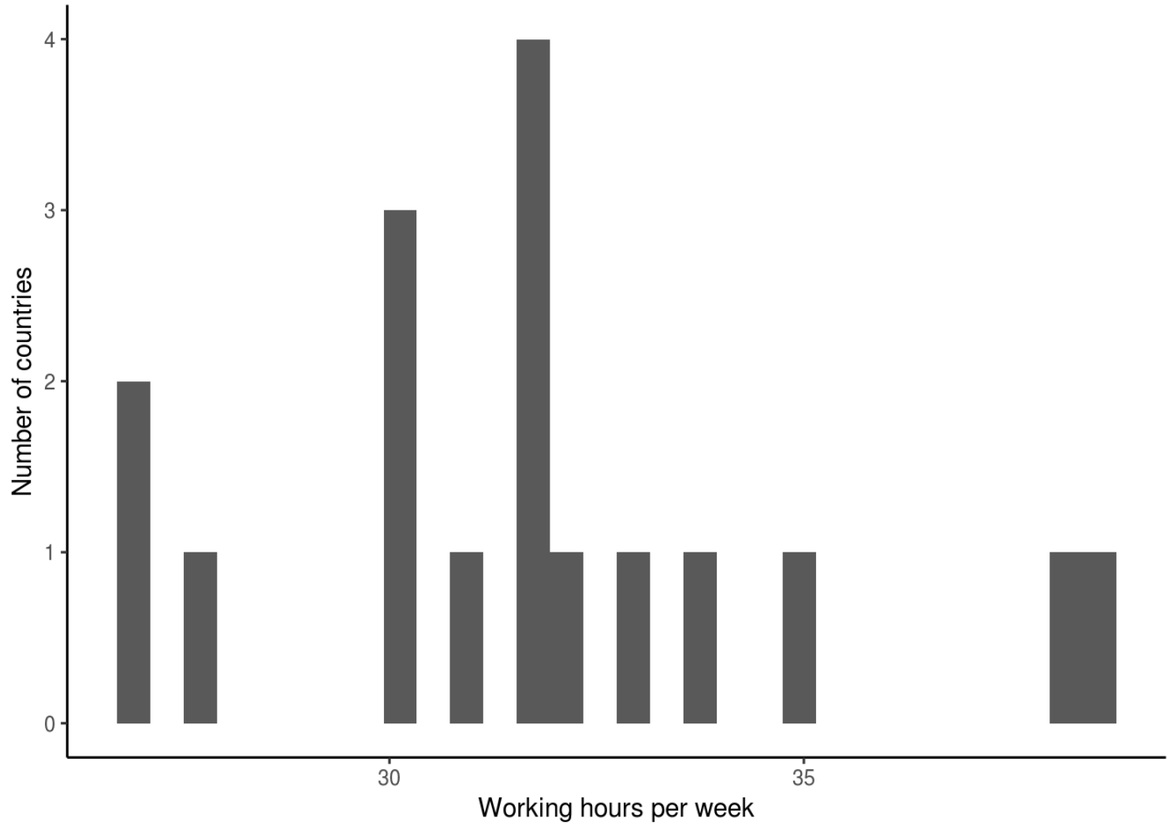
```
  theme(
    text = element_text(
      family = "Bookman",
      color = "gray25")
  )
```



Default ggplot2 themes

```
ggplot(plot_data) +
  geom_histogram(aes(
    x = working_hours)) +
  labs(x = "Working hours per week",
       y = "Number of countries") +
```

```
  theme_classic()
```

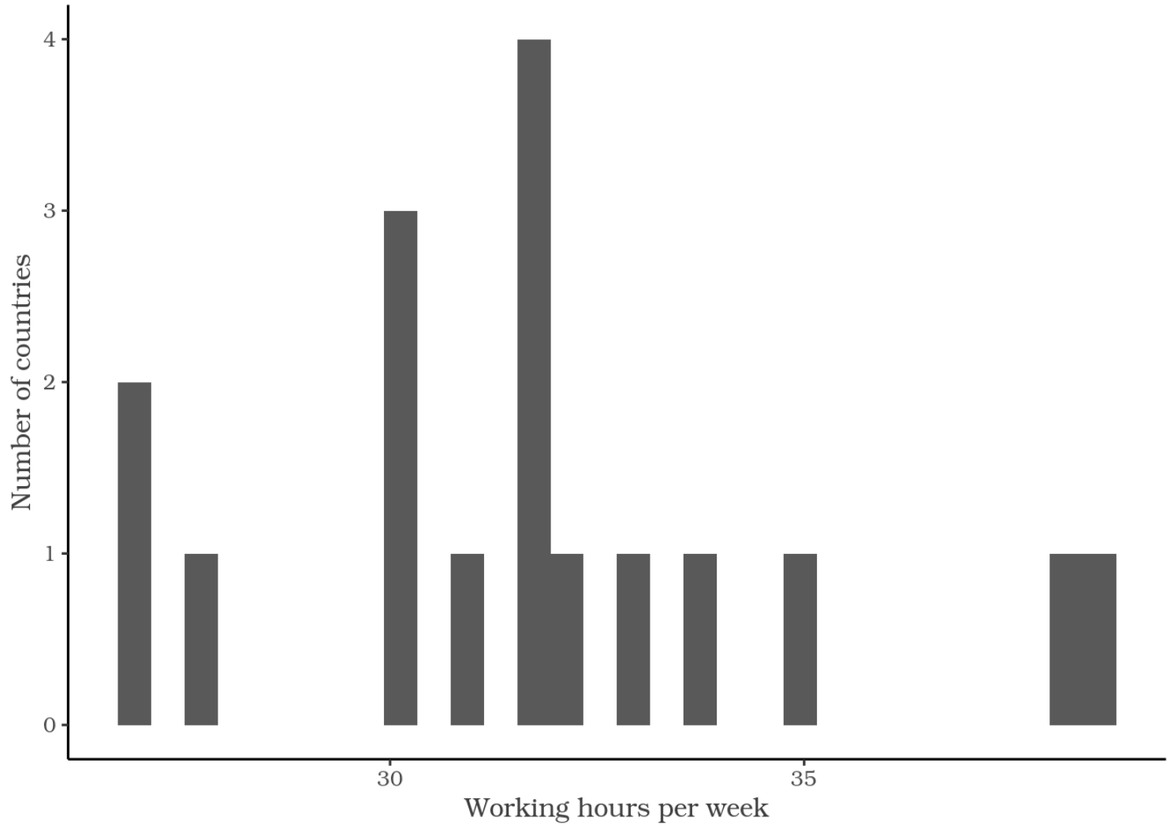


Chaining theme() calls

```
ggplot(plot_data) +
  geom_histogram(aes(
    x = working_hours)) +
  labs(x = "Working hours per week",
       y = "Number of countries") +
```

```
  theme_classic() +
```

```
  theme(
    text = element_text(
      family = "Bookman",
      color = "gray25")
  )
```



Theme configuration options

```
?theme
```

axis.title

label of axes (element_text; inherits from text)

axis.title.x

x axis label (element_text; inherits from axis.title)

axis.title.x.top

x axis label on top axis (element_text; inherits from axis.title.x)

axis.title.x.bottom

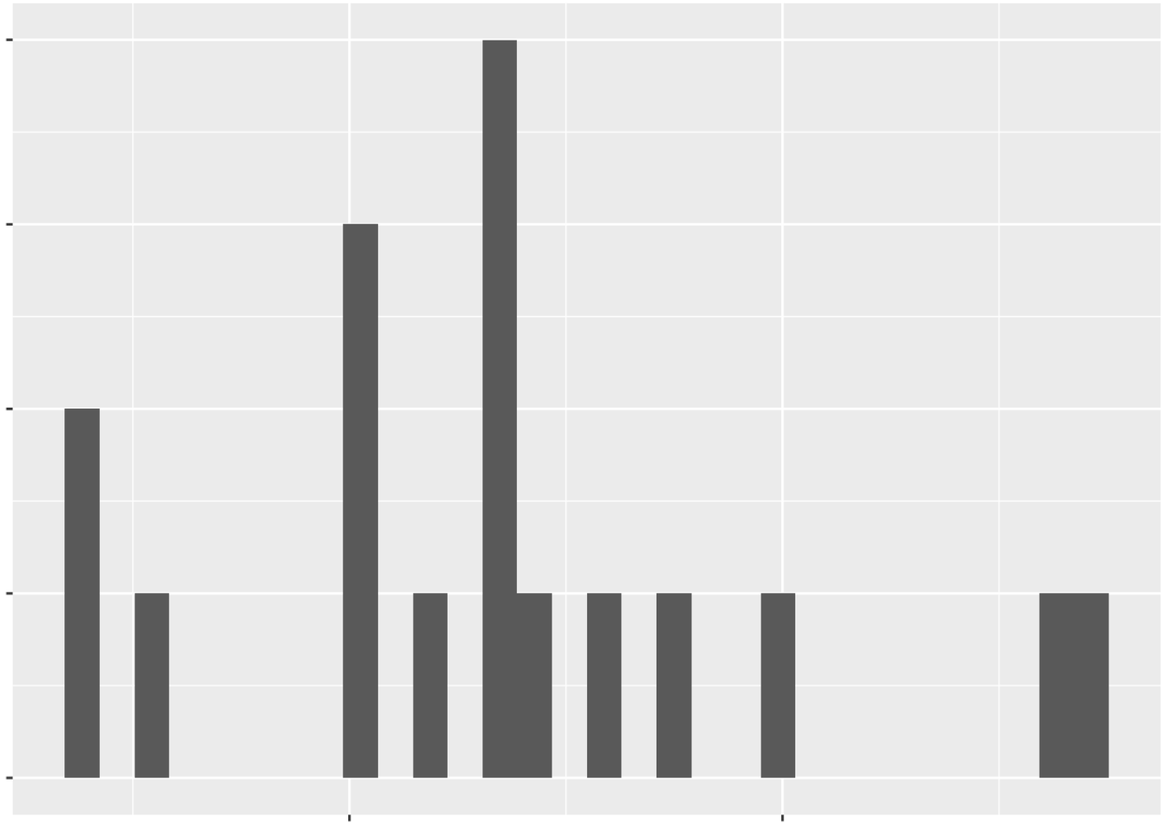
x axis label on bottom axis (element_text; inherits from axis.title.x)

The element_* function family

```
element_text()
element_rect()
element_line()
element_blank()
```

```
ggplot(plot_data) +
  geom_histogram(aes(
    x = working_hours)) +
  labs(x = "Working hours per week",
       y = "Number of countries") +
```

```
  theme(
    text = element_blank()
  )
```



Let's try out themes!

COMMUNICATING WITH DATA IN THE TIDYVERSE