Billionaires broken down

Silvie Cinková

2025-08-09

```
  library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to bec
```

```
  billionaires_df <- read_tsv("../DATA.NPFL112/billionaires_combined.tsv")
```

```
Rows: 28986 Columns: 21
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr (17): person, name.x, state, headquarters, source, industry, gender, las...
dbl  (4): time, daily_income, age, birth_comb

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  billionaires_2020 <- billionaires_df %>%
    filter(time == 2020) %>% select(daily_income, person, world_6region)
```

Billionaires in 2020

```
  glimpse(billionaires_2020)
```

```
Rows: 2,462
Columns: 3
$ daily_income  <dbl> 186317, 408707, 164430, 286074, 124167, 314564, 2575542,~
$ person        <chr> "abdulla_al_futtaim", "abdulla_bin_ahmad_al_ghurair", "a~
$ world_6region <chr> "middle_east_north_africa", "middle_east_north_africa", ~
```

The task performed by the code

- Filter billionaires in 2020
- Boxplots for world regions
- Label outliers with person IDs

The data set `billionaires_df` compiles charts of world's billionaires between 2002 and 2020. This means that the same person can occur more than one time, but their details may differ.

We want to plot billionaires in each world region (column `world_6region`) in 2020 (hence, each person appears only once).

Use boxplots to plot the distribution of `daily_income` among all billionaires in the given world region and text labels with values from the column `person` for outliers. You may have to plot each region separately, that is, not break one plot into facets but run the/a plotting script individually for each world region.

Solution

- Identify outliers
- Create a data frame of outliers
- Plot the boxplot of the entire region
- Over that, text plot with names of outliers
- Bypass overlap by using `ggrepel` library
- plot the upper outlier threshold with label

Identify outliers

- group by region
- upper threshold: Q3 + 1.5 IQR
- lower threshold: Q1 - 1.5 IQR

- is the value an outlier?

Outliers in a boxplot: visualized as individual points. Where do the whiskers end and the area of individual points start? Upper outliers' area starts above the threshold computed as the value on the third quartile (Q3) plus 1.5 time the height of the box (when vertical). This dimension is computed as IQR (interquartile range: Q3 minus Q1). Similarly, the area of lower outliers starts below the box: the first quartile Q1 **minus** 1.5 times IQR. Obviously, when there are no outliers, you see no points. Also, the whiskers are only as long as there are data. They can be shorter than the outlier border if the data is not spread enough!!!

Code to identify outliers

```
billionaires_outliers <- billionaires_2020 %>%
  group_by(world_6region) %>%
  mutate(Q3 = quantile(daily_income, 0.75),
         IQR_col = IQR(daily_income),
         outliers_above = Q3 + 1.5 * IQR_col) %>%
  mutate(is_outlier = if_else(condition = daily_income > outliers_above,
                              true = TRUE,
                              false = FALSE)) %>%
  arrange(desc(daily_income), world_6region) %>% ungroup()
```

This adds a column saying of each income value whether or not it is an outlier in the given world region. And we are actually only interested in the upper outliers because there no lower outliers in the data. We will also use the column `outliers_above`, which contains the threshold for the given region.

Vectors of world regions and of upper thresholds

```
world_6region_vec <- distinct(billionaires_outliers, world_6region) %>%
  arrange(world_6region) %>%
  pull()
outliers_above <- distinct(billionaires_outliers, world_6region,
 ↪  outliers_above) %>%
  arrange(world_6region) %>%
  pull()
```

The (alphabetically) first world region

```
(world_region01 <- world_6region_vec[1])
```

```
[1] "america"
```

All Americans among billionaires

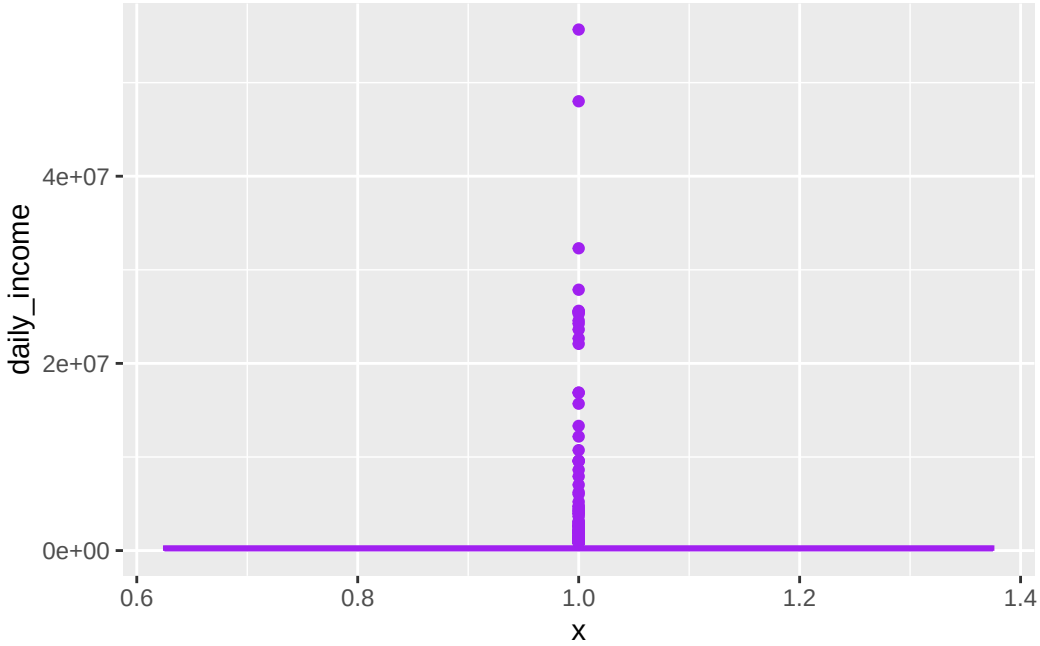
```
all_oneregion <- billionaires_outliers %>%
  filter(world_6region == world_6region_vec[1])
```

All outliers among American billionaires

```
outliers_oneregion <- outliers_oneregion <- all_oneregion %>%
    filter(is_outlier == TRUE)
```

Boxplot of all Americans

```
  ggplot(all_oneregion) + geom_boxplot(mapping = aes(y = daily_income, x = 1),
                color = "purple")
```

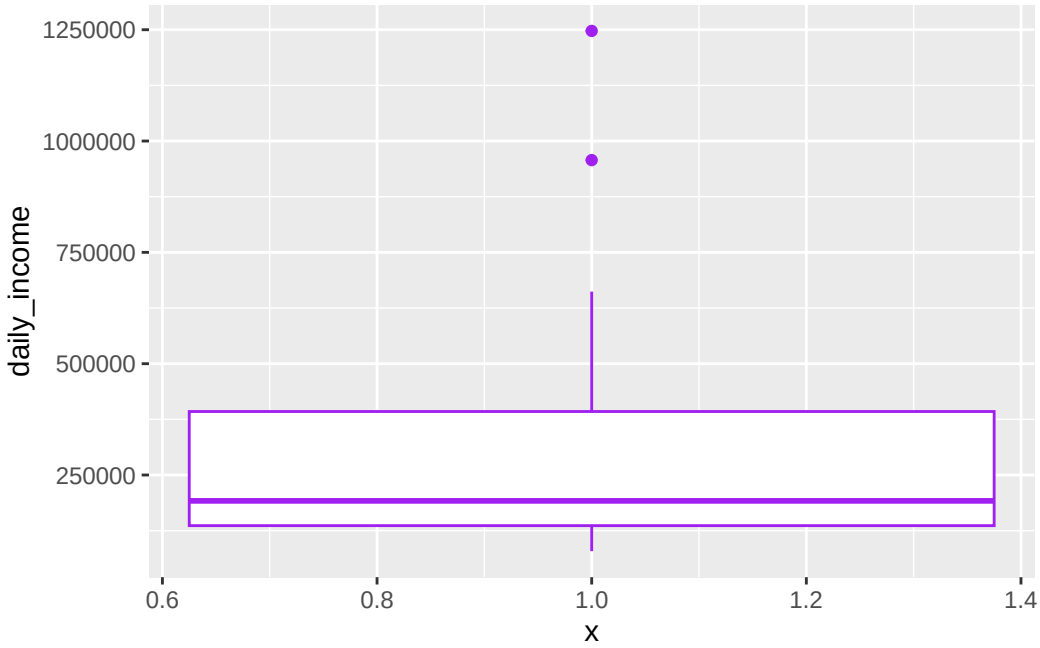


Boxplot of all Africans

```
(world_region04 <- world_6region_vec[4])
```

```
[1] "middle_east_north_africa"
```

```
billionaires_outliers %>%
  filter(world_6region == world_6region_vec[4]) %>%
  ggplot() + geom_boxplot(mapping = aes(y = daily_income, x = 1),
                  color = "purple")
```



We can easily see that the daily incomes of billionaires in different regions vary in scale. Also, American billionaires are so many that we will have to deal with a heavy overplotting.

Plot Americans with outlier labels

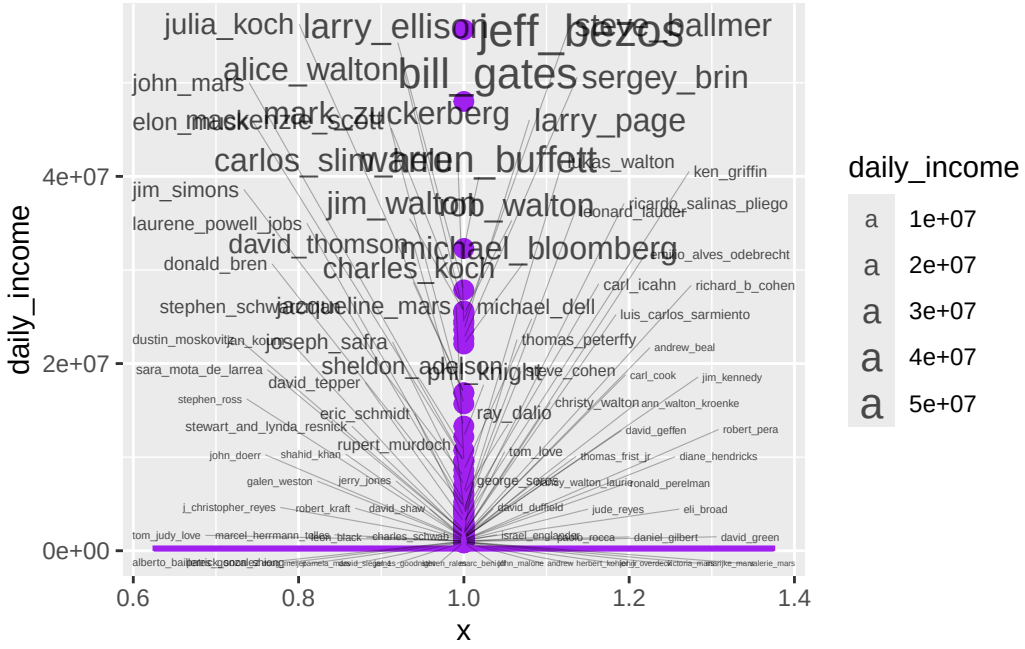
```
library(ggrepel)
set.seed(2331) # change and re-plot until you are happy
 p <- ggplot() +
    geom_boxplot(data = all_oneregion,
```

```
                            mapping = aes(y = daily_income, x = 1),
                            color = "purple", outlier.size = 3)  +
    geom_text_repel(mapping = aes(y = daily_income,
                                  x = 1,
                                  label = person,
                                  size = daily_income),
                    data = outliers_oneregion,
                    max.overlaps = 123,  force = 7,
                    alpha = 0.7,
                    segment.alpha = 0.3,
                    segment.size = 0.2)
```

Plot Americans with outlier labels

```
p
```

```
Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

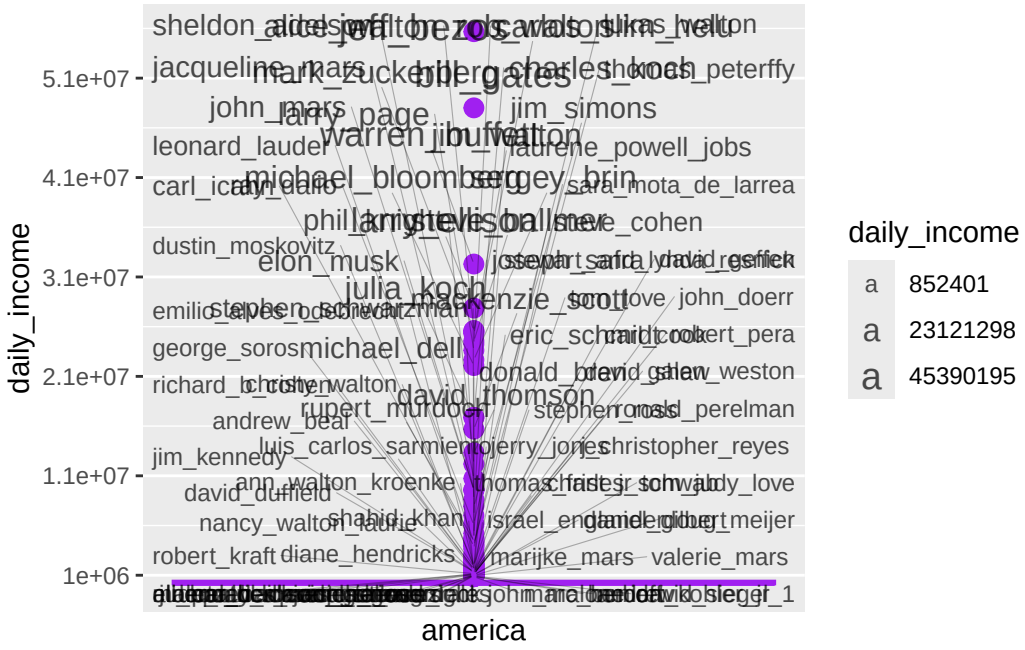


Prettify the plot

```
prettified_01 <- p +
    scale_x_continuous(breaks = NULL,
                       name = world_6region_vec[1]) +
    scale_size_continuous(range = c(3,5),
                          breaks = seq(from =
 ↪  round(min(outliers_oneregion$daily_income)),
                                       to =
 ↪  round(max(outliers_oneregion$daily_income)),
                                       by =
 ↪  round(max(outliers_oneregion$daily_income) * 0.4))) +
    scale_y_continuous(breaks = seq(from = 10^6,
                                    to = ceiling(round(
                                      max(all_oneregion$daily_income) *
 ↪  10^(-6)) * 10^6),
                                    by = 10^7),
                       labels = as.character(seq(from = 10^6,
                                                 to = ceiling(round(

 ↪  max(all_oneregion$daily_income) * 10^(-6)) * 10^6),
                                                 by = 10^7)))
prettified_01
```

```
Warning: ggrepel: 14 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```




```
ggsave("images_ATRIUM/prettified_billionaires01.pdf", scale = 2, limitsize =
↪  FALSE)
```

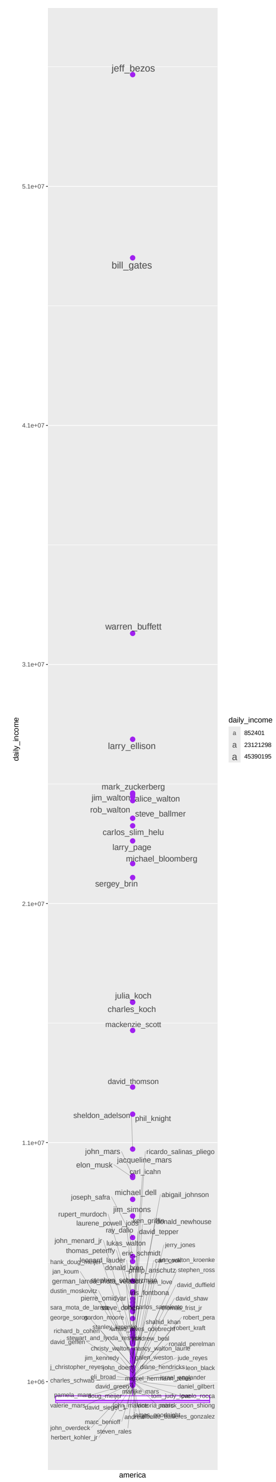
```
Saving 11 x 7 in image
```

The preview in html is probably going to be useless, but look at the output file. Nevertheless: note that the x-axis now says just *america*, with no nonsensical numeric values, and the ticks on the y-axis are all incremented by 10,000,000, starting at the next round value at the upper edge of the box. Also, the size legend has fewer steps (maybe this is not even an improvement - a matter of taste).

A side note: we were taught that we should not map one variable on several aesthetic scales. But I think here it is justified. The names are extremely many and `ggrepel` had to add a lot of jitter. Perhaps size grading is helpful if you want to see on the first glance who the really big animals are.

The prettified plot

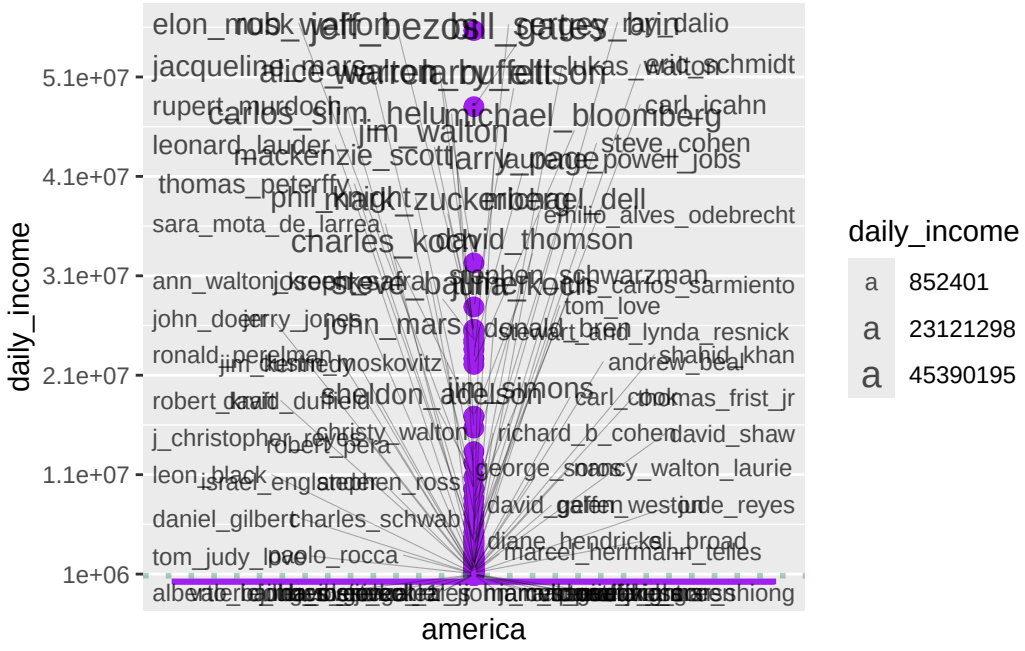
```
prettified_01 # renders poor in slides
```


daily_income
5.1e+07
4.1e+07
3.1e+07
2.1e+07
1.1e+07
1e+06
america
852401
23121298
45390195
jeff_bezos
bill_gates
warren_buffett
larry_ellison
mark_zuckerberg
jim_walton
alice_walton
rob_walton
steve_ballmer
carlos_slim_helu
larry_page
michael_bloomberg
sergey_brin
julia_koch
charles_koch
mackenzie_scott
david_thomson
sheldon_adelson
phil_knight
john_mars
ricardo_salinas_pliego
jacqueline_mars
elon_musk
carl_icahn
michael_dell
abigail_johnson
joseph_safra
jim_simons

Add a line where outliers start

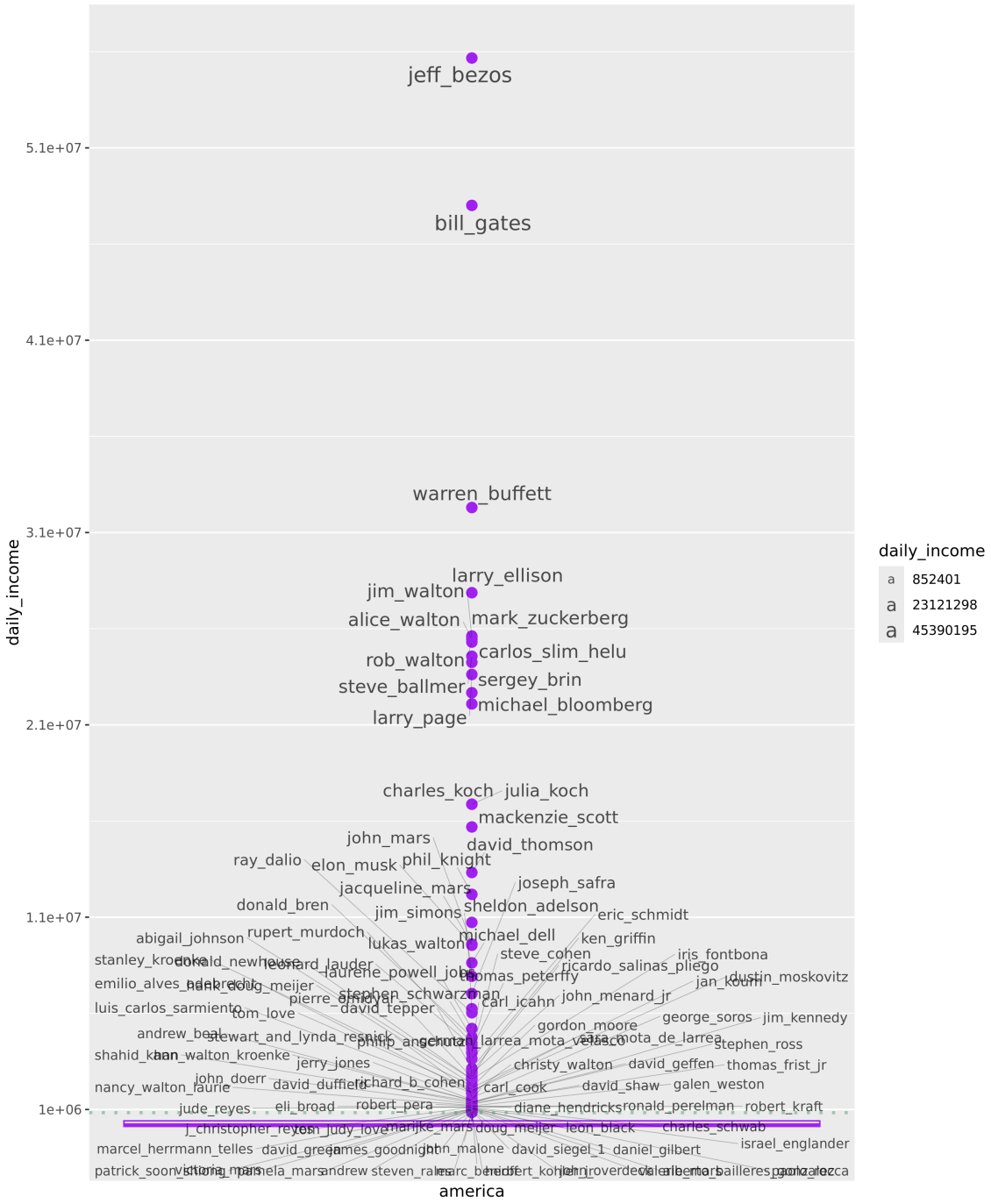
```
prettified_02 <- prettified_01 +  geom_hline(yintercept = outliers_above[1],
                 color = "seagreen",
                 linewidth = 1,
                 linetype=3,
                 alpha = 0.4)
prettified_02
```

```
Warning: ggrepel: 14 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Not pretty here at all, I had to include as saved png file

Prettified plot (from file)



Label the value at which observations become outliers

```
y_axis_offset_for_outlier_label <- 2700000 # manual adjustment
prettified_02 + annotate(geom = "text", x = 0.6,
              y = outliers_above[1] + y_axis_offset_for_outlier_label,
              label = outliers_above[1],
              color = "seagreen",
              size = 6,
              alpha = 0.7)
```

```
Warning: ggrepel: 14 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

Make the position of the label adjustable to different plot scales

```
y_axis_offset_for_outlier_label <- range(all_oneregion$daily_income) %>%
    diff()
  y_axis_offset_for_outlier_label <- y_axis_offset_for_outlier_label * 0.05
```

Offset: a value by which you nudge the label upwards from the line. With one plot, we can do it manually using an absolute value. When we plot the regions in a loop, we need a

relative position towards the line. How about five per cent of the range of the Y-axis? `diff` gives you the difference between minimum and maximum. In the next step, I multiply the value with 0.05. That's 5% of the range. So the distance of the green label from the green line is going to be the same in all plots.

The `if else` condition

One of the plots contained no outliers. The condition says that, in case there are no outliers, we want all billionaire names plotted instead.

Whole code

```
library(ggrepel)
billionaires_outliers <- billionaires_2020 %>%
  group_by(world_6region) %>%
  mutate(Q3 = quantile(daily_income, 0.75),
         IQR_col = IQR(daily_income),
         outliers_above = Q3 + 1.5 * IQR_col) %>%
  mutate(is_outlier = if_else(condition = daily_income > outliers_above,
                              true = TRUE,
                              false = FALSE)) %>%
  arrange(desc(daily_income), world_6region)


world_6region_vec <- distinct(billionaires_outliers, world_6region) %>%
  arrange(world_6region) %>%
  pull()
outliers_above <- distinct(billionaires_outliers, world_6region,
 ↪  outliers_above) %>%
  arrange(world_6region) %>%
  pull()

for (i in seq_along(world_6region_vec)) {
  all_oneregion <- billionaires_outliers %>%
    filter(world_6region == world_6region_vec[i])
  outliers_oneregion <- all_oneregion %>%
    filter(is_outlier == TRUE)
  y_axis_offset_for_outlier_label <- range(all_oneregion$daily_income) %>%
    diff()
  y_axis_offset_for_outlier_label <- y_axis_offset_for_outlier_label * 0.05
  if (nrow(outliers_oneregion) == 0){
    cat(world_6region_vec[i], "has no outliers. I will plot all names.\n")
  p <- ggplot() +
    geom_boxplot(data = all_oneregion,
                 mapping = aes(y = daily_income, x = 1),
                 color = "purple")   +
    geom_text_repel(mapping = aes(y = daily_income,
```

```
                                    x = 1,
                                    label = person,
                                  size = daily_income),
                      data = all_oneregion,
                      max.overlaps = 100,
                      force = 7,
                      alpha = 0.7,
                      segment.alpha = 0.2,
                      segment.size = 0.1) +
    scale_x_continuous(breaks = NULL,
                       name = world_6region_vec[i]) +
    scale_y_continuous(breaks = seq(from = 10^6,
                                    to = ceiling(round(
                                    max(all_oneregion$daily_income) * 10^(-6))
↪   * 10^6),
                                    by = 10^7),
                       labels = as.character(
                                           seq(from = 10^6,
                                               to =
↪   ceiling(round(max(all_oneregion$daily_income) * 10^(-6)) * 10^6),
                                               by = 10^7)
                                    )
                       )
  ggsave(plot = p, filename =
↪   paste0("../my_output_files/outliers_billionaires_", world_6region_vec[i],
↪   ".pdf"),
         width = 7 * 2)

  } else {
  set.seed(155)
  p <- ggplot() +
    geom_boxplot(data = all_oneregion,
                          mapping = aes(y = daily_income, x = 1),
                          color = "purple", outlier.size = 3)  +
    geom_text_repel(mapping = aes(y = daily_income,
                                  x = 1,
                                  label = person,
                                  size = daily_income),
                    data = outliers_oneregion,
                    max.overlaps = 100,  force = 7,
                    alpha = 0.7,
                    segment.alpha = 0.3,
                    segment.size = 0.2) +
    scale_x_continuous(breaks = NULL,
                       name = world_6region_vec[i]) +
    scale_size_continuous(range = c(6,10),
```

```
                            breaks = seq(from =
  ↪ round(min(outliers_oneregion$daily_income)),
                                         to =
  ↪ round(max(outliers_oneregion$daily_income)),
                                         by =
  ↪ round(max(outliers_oneregion$daily_income) * 0.4))) +
    scale_y_continuous(breaks = seq(from = 10^6,
                                    to = ceiling(round(
                                      max(all_oneregion$daily_income) *
  ↪ 10^(-6)) * 10^6),
                                    by = 10^7),
                       labels = as.character(seq(from = 10^6,
                                                 to = ceiling(round(

  ↪ max(all_oneregion$daily_income) * 10^(-6)) * 10^6),
                                                 by = 10^7))) +
    geom_hline(yintercept = outliers_above[i],
               color = "seagreen",
               linewidth = 1,
               linetype=3,
               alpha = 0.4) +
    annotate(geom = "text",
             x = 0.6,
             y = outliers_above[i] + y_axis_offset_for_outlier_label,
             label = outliers_above[i],
             color = "seagreen",
             size = 6) +
    theme(axis.text = element_text(size = 12),
          axis.title = element_text(size = 14))
  cat(world_6region_vec[i], "\n")
  print(p)
  ggsave(plot = p, filename = paste0(
    "../my_output_files/outliers_billionaires_",
    world_6region_vec[i], ".pdf"),
         width = 7 * 2.2)

  }
}
```