talk06 练习与作业

目录

0.1 练习和作业说明

将相关代码填写入以 "'{r} "' 标志的代码框中，运行并看到正确的结果；

完成后，用工具栏里的"Knit" 按键生成 PDF 文档；

将 PDF 文档改为：`姓名-学号-talk06 作业.pdf`，并提交到老师指定的平台/钉群。

0.2 Talk06 及 talk06-practices 内容回顾

1. tidyr

2. 3 个生信任务的 R 解决方案
3. forcats

0.3 练习与作业：用户验证

请运行以下命令，验证你的用户名。

如你当前用户名不能体现你的真实姓名，请改为拼音后再运行本作业！

```
Sys.info()[["user"]]
```

```
## [1] "lucas"
```

```
Sys.getenv("HOME")
```

```
## [1] "/Users/lucas"
```

0.4 练习与作业 1：tidyr

0.4.1 使用 grades 变量做练习

1. 装入 grades 变量；

library(dplyr);

grades <- read_tsv(file = "data/talk05/grades.txt");

2. 使用 tidyr 包里的 pivot_longer 和 pivot_wider 函数对 grades 变量进行宽长转换；

```
## 代码写这里，并运行；
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(readr)
grades =
  read_tsv(
    file=
      "../data/talk05/grades.txt")
```

```
## Rows: 9 Columns: 3

## -- Column specification ---------------------------------------------------------
## Delimiter: "\t"
## chr (2): name, course
## dbl (1): grade
##
## 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.
```

```r
# Make the data longer
grades_long =
  grades %>%
    group_by(name, course) %>%
    summarize(
      grade = mean(
        as.numeric(grade),
        na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'name'. You can override using the
## `.groups` argument.
```

```
# Make the data wider
grades_wide =
  grades %>%
  pivot_wider(
    names_from = course,
    values_from = grade)

# Print the data
print(grades_long)
```

```
## # A tibble: 9 x 3
## # Groups:   name [3]
##   name        course         grade
##   <chr>       <chr>          <dbl>
## 1 Kang Ning   Bioinformatics   100
## 2 Kang Ning   Chemistry         76
## 3 Kang Ning   Chinese           20
## 4 Weihua Chen Bioinformatics    99
## 5 Weihua Chen English           99
## 6 Weihua Chen Microbiology      89
## 7 Zhi Liu     Chinese           69
## 8 Zhi Liu     English           50
## 9 Zhi Liu     Microbiology     100
```

```
print(grades_wide)
```

```
## # A tibble: 3 x 6
##   name        Microbiology English Chinese Bioinformatics Chemistry
##   <chr>              <dbl>   <dbl>   <dbl>          <dbl>     <dbl>
## 1 Zhi Liu              100      50      69             NA        NA
## 2 Weihua Chen           89      99      NA             99        NA
## 3 Kang Ning             NA      NA      20            100        76
```

3. 使用 pivot_longer 时，有时会产生 na 值，如何使用此函数的参数去除带 na 的行？

```
## 代码写这里，并运行；
library(tidyr)

grades_long2 =
  grades %>%
    group_by(name, course) %>%
    summarize(
      grade = mean(
        as.numeric(grade),
        na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'name'. You can override using the
## `.groups` argument.
```

```
print(grades_long2)
```

```
## # A tibble: 9 x 3
## # Groups:   name [3]
##   name        course         grade
##   <chr>       <chr>          <dbl>
## 1 Kang Ning   Bioinformatics   100
## 2 Kang Ning   Chemistry         76
## 3 Kang Ning   Chinese           20
## 4 Weihua Chen Bioinformatics    99
## 5 Weihua Chen English           99
## 6 Weihua Chen Microbiology      89
## 7 Zhi Liu     Chinese           69
## 8 Zhi Liu     English           50
## 9 Zhi Liu     Microbiology     100
```

4. 以下代码有什么作用?

```
grades %>% complete( name, course )
```

答:

这段代码使用 `dplyr` 和 `tidyr` 中的 `complete()` 函数来生成一个完整的数据框，确保每个不同的 name 和 course 组合都存在，并且如果数据中没有这些组合，则填充缺失的行。这通常用于数据填充和确保数据表格中包含所有可能的组合。

0.5 练习与作业 2：作图

0.5.1 用下面的数据作图

1. 利用下面代码读取一个样本的宏基因组相对丰度数据

```
abu <-
  read_delim(
    file = "../data/talk06/relative_abundance_for_RUN_ERR1072629_taxonlevel_species.txt
    delim = "\t", quote = "", comment = "#");
```

2. 取前 5 个丰度最高的菌，将其它的相对丰度相加并归为一类 `Qita`;
3. 用得到的数据画如下的空心 pie chart:

```
## 代码写这里，并运行；
# Load the library
library(readr)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

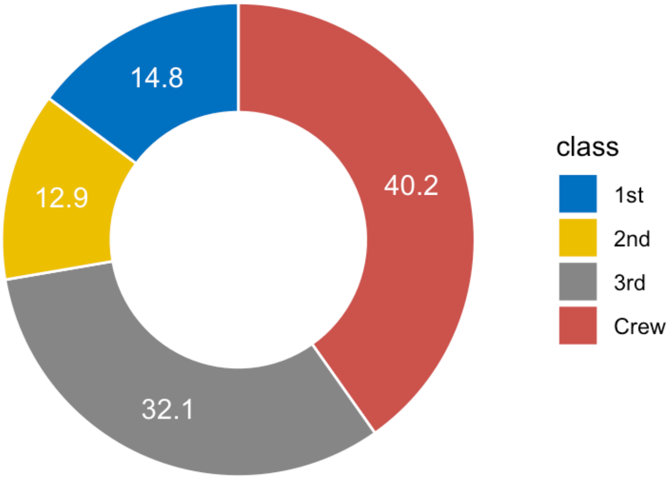


图 1: make a pie chart like this using the meteagenomics data

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Read the data
abu =
  read_delim(
    file =
      "../data/talk06/relative_abundance_for_RUN_ERR1072629_taxonlevel_species.txt",
    delim = "\t",
    quote = "",
    comment = "#");
```

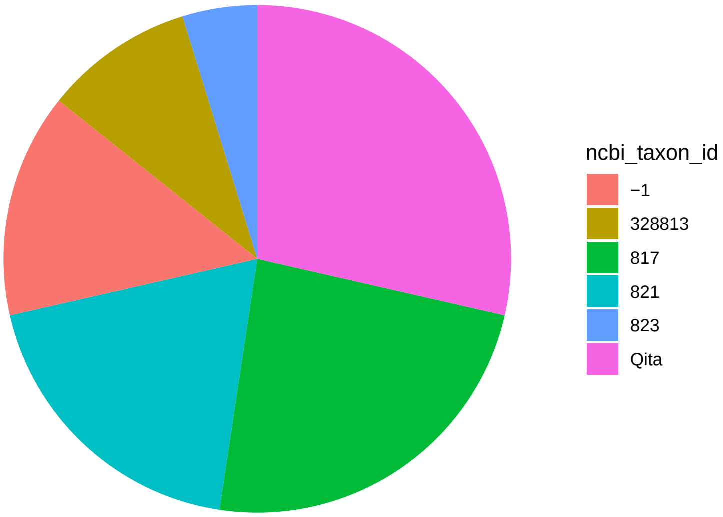
```
## Rows: 122 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): scientific_name
## dbl (2): ncbi_taxon_id, relative_abundance
##
## 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.
```

```r
# Re-order the data
abu_reorder =
  abu[order(-abu$relative_abundance), ]

# Find out top 5
abu_top5 =
  abu_reorder[1:5, ]

# Add the rest of the data
abu_qita =
  sum(abu[6:nrow(abu_reorder), ]$relative_abundance)
```

```
# Combine them
abu_full =
  rbind(abu_top5,c("Qita", abu_qita))
```

```
## Warning in rbind(deparse.level, ...): number of columns of result, 3, is not a
## multiple of vector length 2 of arg 2
```

```
# Draw the plot
abu_pie_chart =
  ggplot(data = abu_full,
         aes(x = "",
             y = relative_abundance,
             fill = ncbi_taxon_id)) +
  geom_bar(stat = "Identity", width = 1) +
  coord_polar(theta = "y") +
  theme_void() +
  theme(legend.position = "right")

# Print the pie chart
print(abu_pie_chart)
```

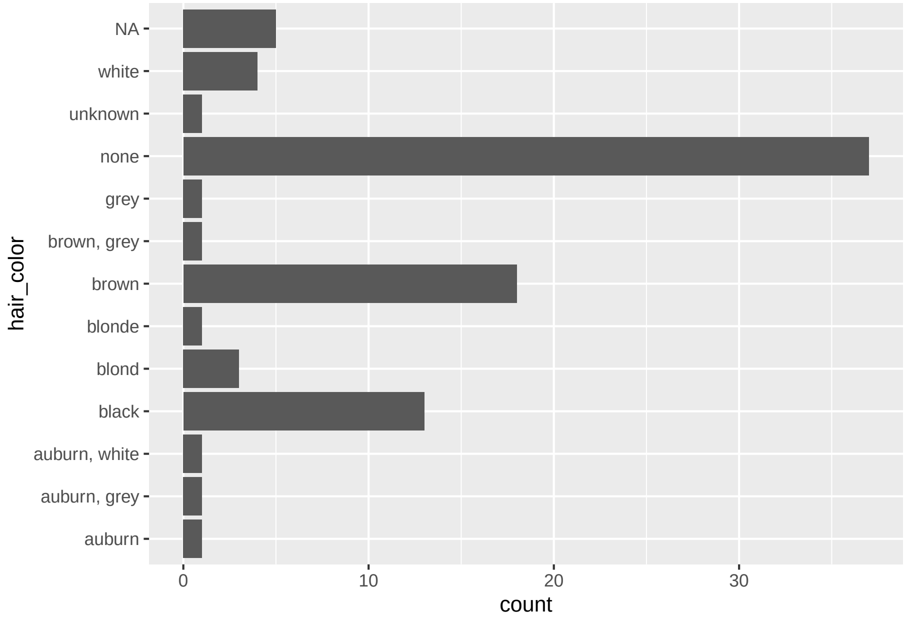


0.5.2 使用 starwars 变量做图

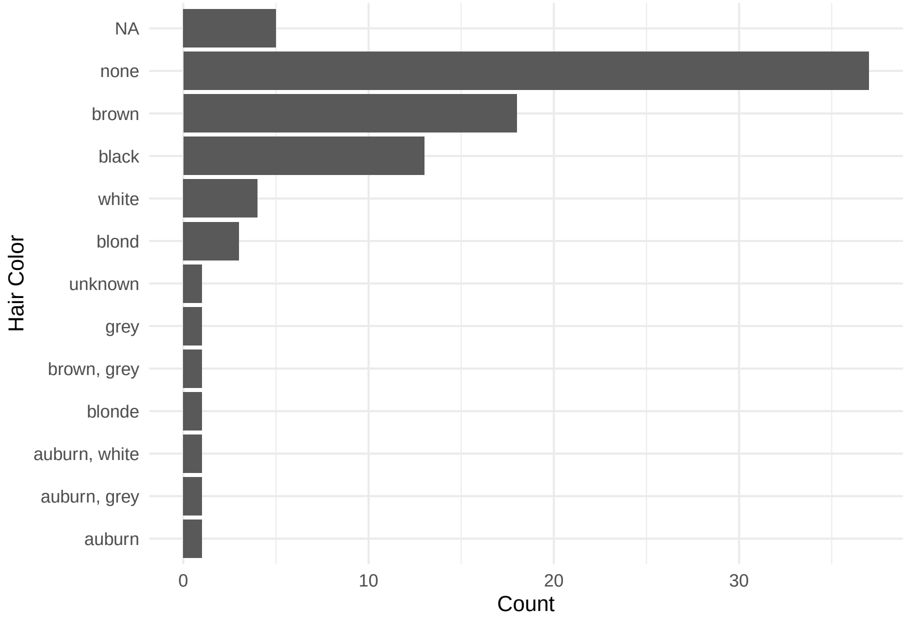
1. 统计 starwars 中 hair_color 的种类与人数时，可用下面的代码:

但是，怎么做到**按数量从小到大**排序?

```r
library(dplyr)
library(ggplot2)
library(forcats)
ggplot(starwars, aes(x = hair_color)) +
  geom_bar() +
  coord_flip()
```



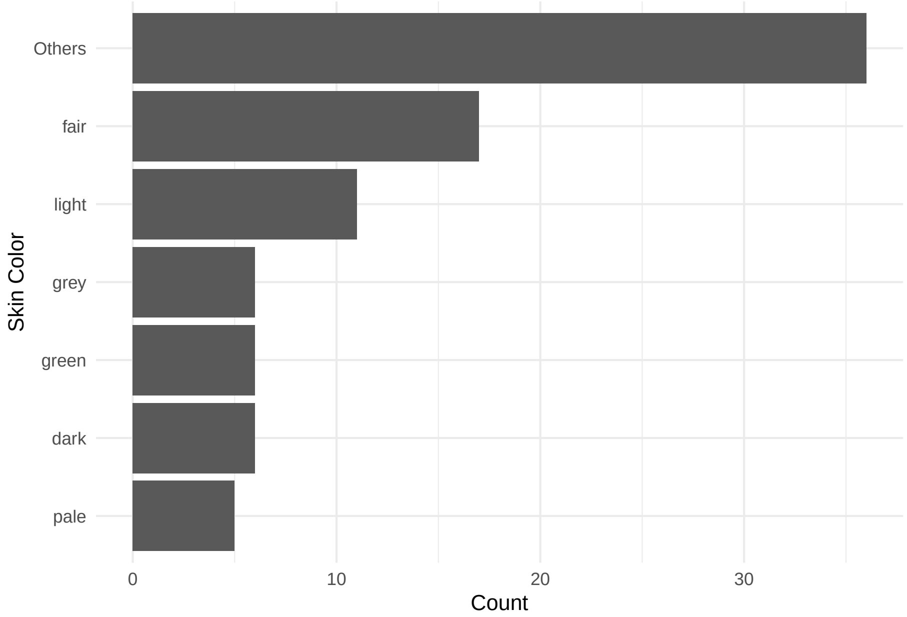
```
## 代码写这里，并运行；
library(dplyr)

# Count the number o
# and sort them in descending order.
starwars_counts =
  starwars %>%
  group_by(hair_color) %>%
  summarise(count = n()) %>%
  arrange(count)

# Draw the plot
starwars_plot01=
  ggplot(
    starwars_counts,
    aes(
      x = reorder(
        hair_color,
        count),
      y = count)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(
    x = "Hair Color",
    y = "Count") +
  theme_minimal()

# Print the plot
print(starwars_plot01)
```



2. 统计 `skin_color` 时，将出现频率小于 0.05（即 5%）的颜色归为一类 `Others`，按出现次数排序后，做与上面类似的 barplot；

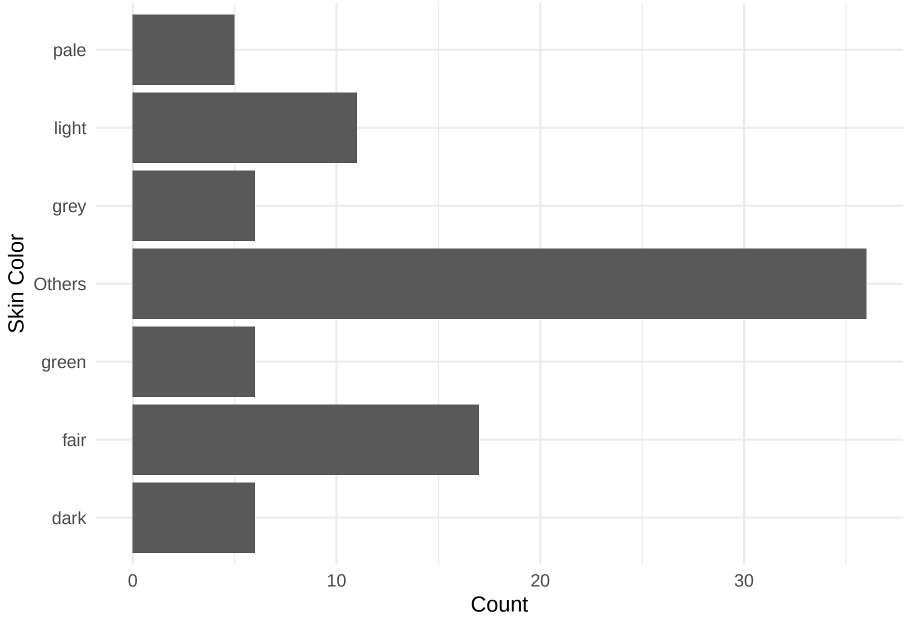
```
## 代码写这里，并运行；
# Count the number of
# skin_color types and people
```

```
starwars_counts_skin =
  starwars %>%
  group_by(skin_color) %>%
  summarise(count = n())

# Group the category 'Others'
total_count_group =
  sum(
    starwars_counts_skin$count)
threshold =
  0.05 * total_count_group

starwars_counts_skin_group =
  starwars_counts_skin %>%
  mutate(
    skin_color =
      ifelse(
        count < threshold,
        "Others",
        as.character(skin_color)))

# Recount the type and number.
starwars_counts_skin_group2 =
  starwars_counts_skin_group %>%
  group_by(skin_color) %>%
  summarise(count = sum(count)) %>%
  arrange(count)

# Draw the plot
starwars_plot02 =
  ggplot(
    starwars_counts_skin_group2,
    aes(
      x = reorder(
        skin_color,
        count),
      y = count)) +
  geom_bar(
    stat = "identity") +
  coord_flip() +
  labs(
    x = "Skin Color",
    y = "Count") +
  theme_minimal()

# Print the plot
print(starwars_plot02)
```



3. 使用 2 的统计结果，但画图时，调整 bar 的顺序，使得 `Others` 处于第 4 的位置上。提示，可使用 `fct_relevel` 函数；

```r
## 代码写这里，并运行；
# Load the library
library(forcats)

# Count the number
starwars_counts_skin2 =
  starwars %>%
  group_by(skin_color) %>%
  summarise(count = n())

# Group the category 'Others'
total_count_group2 =
  sum(starwars_counts_skin2$count)
threshold =
  0.05 * total_count_group2

starwars_counts_skin2_group =
  starwars_counts_skin2 %>%
  mutate(
    skin_color =
      ifelse(
        count < threshold,
        "Others",
        as.character(skin_color)))

# Recount the type and number.
starwars_counts_skin2_group2 =
  starwars_counts_skin2_group %>%
  group_by(skin_color) %>%
  summarise(count = sum(count))

# Adjusting the position of 'Others'
starwars_counts_skin2_group2$skin_color =
  fct_relevel(
  starwars_counts_skin2_group2$skin_color,
  "Others",
  after = 3  # to 4th
)

# Draw the plot
starwars_plot03 =
  ggplot(
    starwars_counts_skin2_group2,
    aes(
      x = skin_color,
      y = count)) +
  geom_bar(
    stat = "identity") +
  labs(
    x = "Skin Color",
    y = "Count") +
  coord_flip() +
  theme_minimal()

# Print the plot
print(starwars_plot03)
```



0.6 练习与作业 3：数据分析

0.6.1 使用 STRING PPI 数据分析并作图

1. 使用以下代码，装入 PPI 数据；

```
ppi <- read_delim( file = "../data/talk06/ppi900.txt.gz", col_names = T,
                   delim =  "\t", quote = "" );
```

2. **随机挑选**一个基因，得到类似于本章第一部分的互作网络图；

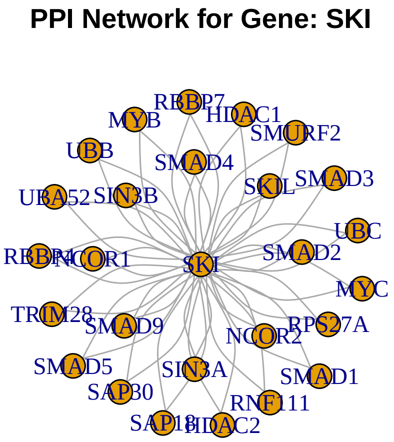
```
## 代码写这里，并运行；
# Load the packages
library(readr)
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following object is masked from 'package:tidyr':
##
##     crossing

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```r
library(dplyr)

# 1. Load PPI data
ppi =
  read_delim(
    file = "../data/talk06/ppi900.txt.gz",
    col_names = TRUE,
    delim = "\t",
    quote = "")
```

```
## Rows: 504436 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): gene1, gene2
## dbl (1): score
##
## 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.
```

```r
# 2. Randomly select a gene
set.seed(123)
# Setting random seeds to
# ensure reproducible results
random_gene =
  ppi %>%
    pull(gene1) %>%
    sample(1)

# 3. Creating Interaction Network Diagrams
ppi_network =
  ppi %>%
    filter(gene1 == random_gene | gene2 == random_gene) %>%
    select(gene1, gene2) %>%
    unique() %>%
    graph.data.frame(directed = FALSE)

# Mapping of Interaction Networks
plot_ppi =
  plot(
    ppi_network,
    layout = layout.auto(ppi_network),
    main = paste(
      "PPI Network for Gene:",
      random_gene))
```



```r
# Print the plot
print(plot_ppi)
```

```
## NULL
```

0.6.2 对宏基因组相对丰度数据进行分析

1.data/talk06 目录下有 6 个文本文件，每个包含了一个宏基因组样本的分析结果：

```
relative_abundance_for_curated_sample_PRJEB6070-DE-073_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-074_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-075_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-076_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-077_at_taxonlevel_species.txt
```

2. 分别读取以上文件，提取 `scientific_name` 和 `relative_abundance` 两列；

3. 添加一列为样本名，比如 PRJEB6070-DE-073, PRJEB6070-DE-074 … ;
4. 以 scientific_name 为 key，将其内容合并为一个 data.frame 或 tibble，其中每行为一个样本，每列为样本的物种相对丰度。注意：用 join 或者 spread 都可以，只要能解决问题。
5. 将 NA 值改为 0。

```r
## 代码写这里，并运行；
# Load the required packages
library(dplyr)
library(tidyr)
library(readr)

# Creating a list of file paths
file_list = list(
  "data/talk06/relative_abundance_for_curated_sample_PRJEB6070-DE-073_at_taxonlevel_spe
  "data/talk06/relative_abundance_for_curated_sample_PRJEB6070-DE-074_at_taxonlevel_spe
  "data/talk06/relative_abundance_for_curated_sample_PRJEB6070-DE-075_at_taxonlevel_spe
  "data/talk06/relative_abundance_for_curated_sample_PRJEB6070-DE-076_at_taxonlevel_spe
  "data/talk06/relative_abundance_for_curated_sample_PRJEB6070-DE-077_at_taxonlevel_spe
)

# Read all the files,
# skip the first three lines,
# and merge them into one dataframe
all_data =
  lapply(file_list, function(file_path) {
    sample_data =
      read_delim(
        file_path,
        delim = "\t",
        skip = 3,
        show_col_types = FALSE)
    sample_name =
      gsub(".*sample_(.*?)_at_taxonlevel.*",
           "\\1",
           basename(file_path))
    sample_data$sample_name =
      sample_name
    return(sample_data)
  }) %>%
  bind_rows()

# Merge the rows
result_df =
  all_data %>%
  # Remove unneeded columns
    select(
      -ncbi_taxon_id,
      -taxon_rank_level) %>%
    group_by(
      sample_name,
      scientific_name) %>%
    summarize(
      relative_abundance =
        sum(
          relative_abundance))
```

```
## `summarise()` has grouped output by 'sample_name'. You can override using the
## `.groups` argument.
```

```
# Change NA value to 0
result_df[
  is.na(
    result_df)] =
      0
```

```r
# Print the result
head(result_df)
```

```
## # A tibble: 6 x 3
## # Groups:   sample_name [1]
##   sample_name      scientific_name             relative_abundance
##   <chr>            <chr>                                    <dbl>
## 1 PRJEB6070-DE-073 Adlercreutzia equolifaciens            0.656
## 2 PRJEB6070-DE-073 Alistipes finegoldii                   0.307
## 3 PRJEB6070-DE-073 Alistipes onderdonkii                  2.59
## 4 PRJEB6070-DE-073 Alistipes sp. HGB5                     2.04
## 5 PRJEB6070-DE-073 Anaerostipes hadrus                    0.0179
## 6 PRJEB6070-DE-073 Anaerotruncus colihominis              0.00083
```