Comparison of savings potentials between measures

Marlene Kindler

2025-03-19

Code for the comparison of savings potentials between measures.

To execute, change:
- Login data for Nextcloud
- Adjust download location
- Adjust upload location
- Adjust URLs for uploading

Load Packages:

```
# install.packages("httr")
# install.packages("readxl")
# install.packages("dplyr")
# install.packages("writexl")
# install.packages("ggplot2")
# install.packages("tidyr")
# install.packages("openxlsx")
# install.packages("flextable")
# install.packages("officer")
# install.packages("tidyverse")
# install.packages("viridis")

library(readxl)
library(httr)
library(dplyr)
library(writexl)
library(ggplot2)
library(tidyr)
library(openxlsx)
library(flextable)
library(officer)
library(tidyverse)
```

```
## Warning: Paket 'tidyverse' wurde unter R Version 4.4.3 erstellt

## Warning: Paket 'readr' wurde unter R Version 4.4.3 erstellt

## Warning: Paket 'forcats' wurde unter R Version 4.4.3 erstellt

## Warning: Paket 'lubridate' wurde unter R Version 4.4.3 erstellt
```

```
library(viridis)
```

```
## Warning: Paket 'viridis' wurde unter R Version 4.4.3 erstellt
```

Parameters

Parameters to be adjusted:

```
# storage location
location_download <- "C:/Users/Klene/Documents/Uni_Bremen/WS24_25/Masterarbei
t/R/Daten/"
location_upload <- "C:/Users/Klene/Documents/Uni_Bremen/WS24_25/Masterarbeit/
R/Results/"
# Login data for NextCloud
username <- "mkindler@uni-bremen.de" # username
password <- "DLCi3-Qc4iD-dcHzR-fJaXr-cyKHR" # password
# WebDAV-URL to upload
nextcloud_url_2 <- "https://nc.uni-bremen.de/remote.php/dav/files/mkindler%40
uni-bremen.de/Masterarbeit/Masterarbeit_II/R/Results/"
```

Data preparation

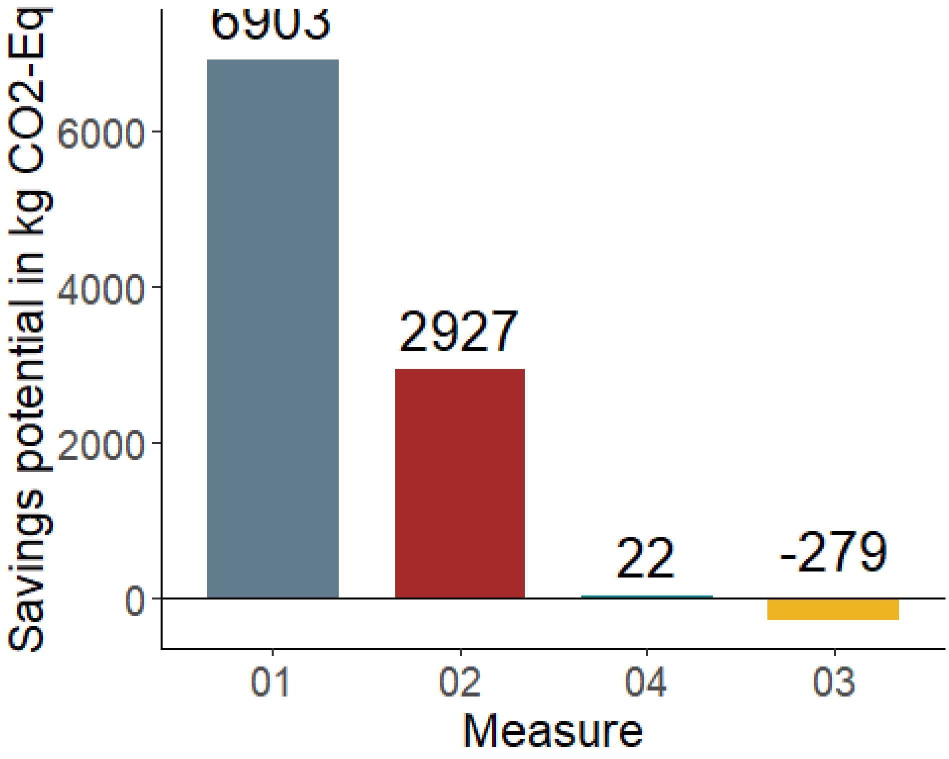
Load data:

```
# Load results data from Excel file
results1 <- read.csv(paste0(location_upload, "01_data.csv"), check.names = FA
LSE, stringsAsFactors = FALSE)
results2 <- read.csv(paste0(location_upload, "02_data.csv"), check.names = FA
LSE, stringsAsFactors = FALSE)
results3 <- read.csv(paste0(location_upload, "03_data.csv"), check.names = FA
LSE, stringsAsFactors = FALSE)
results4 <- read.csv(paste0(location_upload, "04_data.csv"), check.names = FA
LSE, stringsAsFactors = FALSE)

# select only data on Climate Change
climateChange1 <- results1[results1$`impact category` == "Climate change",] %
>%
  select("SP total", "norm. SP total", "prop. SP total", "relevant") %>%
  rename(SP = "SP total",
         "norm. SP" = "norm. SP total",
         "prop. SP" = "prop. SP total") %>%
  mutate(measure = "01")
climateChange2 <- results2[results2$`impact category` == "Climate change",] %
>%
  select("SP FR", "norm. SP FR", "prop. SP FR", "relevant") %>%
  rename(SP = "SP FR",
         "norm. SP" = "norm. SP FR",
         "prop. SP" = "prop. SP FR") %>%
  mutate(measure = "02")
climateChange3 <- results3[results3$`impact category` == "Climate change",] %
>%
  select("SP RF", "norm. SP RF", "prop. SP RF", "relevant") %>%
  rename(SP = "SP RF",
         "norm. SP" = "norm. SP RF",
         "prop. SP" = "prop. SP RF") %>%
  mutate(measure = "03")
climateChange4 <- results4[results4$`impact category` == "Climate change",] %
>%
  select("SP MS", "norm. SP MS", "prop. SP MS", "relevant") %>%
  rename(SP = "SP MS",
         "norm. SP" = "norm. SP MS",
         "prop. SP" = "prop. SP MS") %>%
  mutate(measure = "04")

climateChange <- bind_rows(climateChange1, climateChange2, climateChange3, cl
imateChange4)
```

Plot

Plot the climate change savings potential

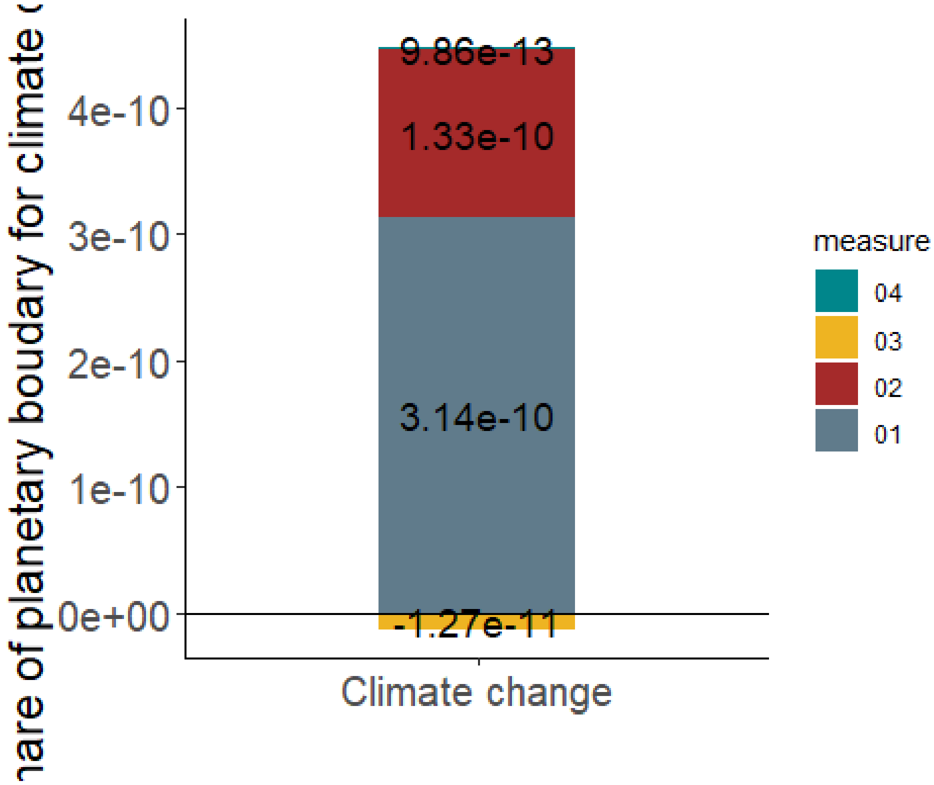
```
climateChange <- climateChange %>%
  arrange(desc(SP))
climateChange$measure <- factor(climateChange$measure, levels = climateChange
$measure)

# Plot Climate change" impact category
ggplot(climateChange, aes(x = `measure`, y = `SP`, fill = `measure`)) +
  geom_bar(stat = "identity", width = 0.7, show.legend = FALSE) +
  geom_hline(yintercept = 0, color = "black", linetype = "solid") +
  geom_text(
    aes(
      label = round(`SP`, 0),
      vjust = ifelse(measure == "03", -1.2, -0.5)
    ),
    size = 7
  ) +
  labs(
    x = "Measure",
    y = "Savings potential in kg CO2-Eq"
  ) +
  scale_fill_manual(values = c("01" = "lightskyblue4",
                               "02" = "brown",
                               "03" = "goldenrod2",
                               "04" = "turquoise4")) +
  theme_classic() +
  theme(
    axis.text.x = element_text(size = 15),
    axis.text.y = element_text(size = 15),
    axis.title.x = element_text(size = 17),
    axis.title.y = element_text(size = 17)
  ) +
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.09)))
```



```
# Save plot
ggsave(paste0(location_upload, "Comparison_CO2.png"), dpi = 600)

## Saving 5 x 4 in image
```

Plot normalized data:

```
climateChange <- climateChange %>%
  mutate(x = "Climate change")

# arrange accending
climateChange <- climateChange %>%
  arrange(abs(`norm. SP`))
climateChange$measure <- factor(climateChange$measure, levels = climateChange
$measure)

ggplot(climateChange, aes(x = x, y = `norm. SP`, fill = `measure`)) +
  geom_bar(position = "stack" , stat = "identity", width = 0.4, show.legend =
TRUE) +
  geom_hline(yintercept = 0, color = "black", linetype = "solid") +
  geom_text(
    aes(label = formatC(`norm. SP`, format = "e", digits = 2)),
    position = position_stack(vjust = 0.5),
    size = 5,
    color = "black"
  ) +
  labs(
    x = "",
    y = "Share of planetary boudary for climate change") +
  scale_fill_manual(values = c("01" = "lightskyblue4",
                               "02" = "brown",
                               "03" = "goldenrod2",
                               "04" = "turquoise4")) +
  theme_classic() +
  theme(
    axis.text.x = element_text(size = 15),
    axis.text.y = element_text(size = 15),
    axis.title.x = element_text(size = 17),
    axis.title.y = element_text(size = 17)
  ) +
  theme(legend.position = "right")
```

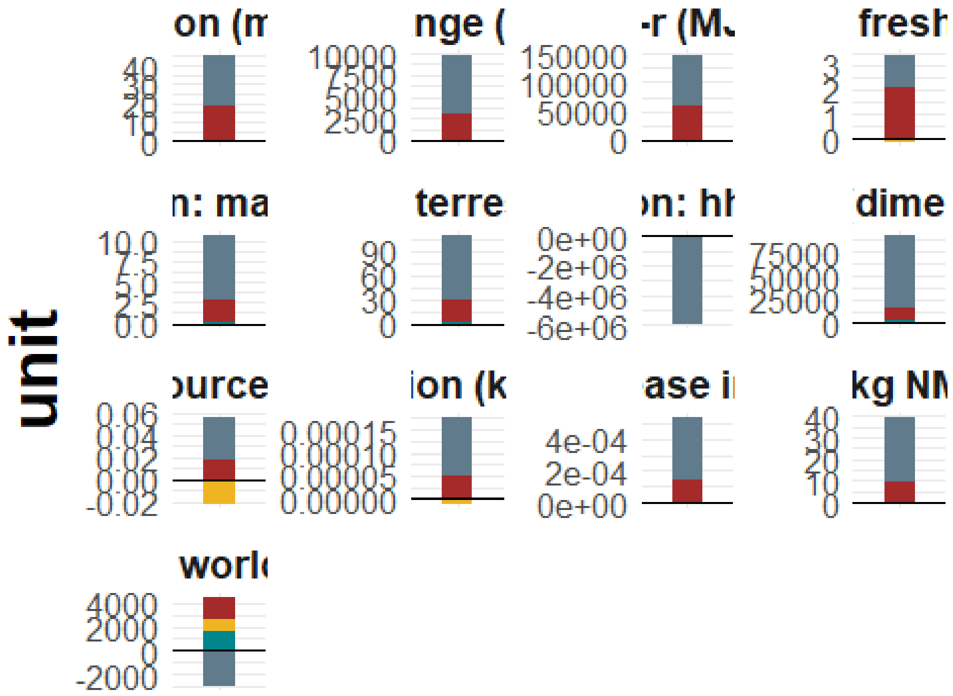


```
# Save plot
ggsave(paste0(location_upload, "Comparison_norm_CO2.png"), dpi = 600)

## Saving 5 x 4 in image
```

Plot all absolute impact categories

```
# Data preparation
data1 <- results1 %>%
  rename("measure 1" = "SP total") %>%
  select("impact category", "unit", "measure 1")
data2 <- results2 %>%
  rename("measure 2" = "SP FR") %>%
  select("impact category", "measure 2")
data3 <- results3 %>%
```

```r
  rename("measure 3" = "SP RF") %>%
  select("impact category", "measure 3")
data4 <- results4 %>%
  rename("measure 4" = "SP MS") %>%
  select("impact category", "measure 4")

data <- left_join(data1, data2, by = c("impact category" = "impact category")
)
data <- left_join(data, data3, by = c("impact category" = "impact category"))
data <- left_join(data, data4, by = c("impact category" = "impact category"))

# Save table of all normalized results:
data_table <- data %>%
  mutate(across(
    .cols = c(`measure 1`, `measure 2`, `measure 3`, `measure 4`),
    .fns = ~ format(.x, scientific = TRUE, digits = 3)
  ))
ft <- flextable(data_table) %>%
  theme_vanilla() %>%
  autofit() %>%
  bold(part = "header") %>%
  border_outer() %>%
  border_inner_h() %>%
  border_inner_v()
# Save as PNG
tf <- tempfile(fileext = ".png")
save_as_image(x = ft, path = paste0(location_upload, "_all_results", ".png"))

## [1] "C:/Users/Klene/Documents/Uni_Bremen/WS24_25/Masterarbeit/R/Results/_a
ll_results.png"

data$`impact category` <- recode(data$`impact category`,
  "Energy resources: non-renewable" = "Energy resources n-r",
  "Ionising radiation: human health" = "Ionising radiation: hh",
  "Material resources: metals/minerals" = "Material resources",
  "Particulate matter formation" = "PMF",
  "Photochemical oxidant formation: human health" = "POF: hh"
)
data <- data %>%
  mutate(`impact category` = paste0(`impact category`, " (", unit, ")"))
data <- data %>%
  select(-unit)

data_long <- data

# Transform data into long format
data_long <- data_long %>%
  gather(key = "measure", value = "value", -`impact category`)


ggplot(data_long, aes(x = `impact category`, y = value, fill = measure)) +
  geom_bar(position = "stack", stat = "identity", width = 0.4, show.legend =
TRUE) +
  geom_hline(yintercept = 0, color = "black", linetype = "solid") +
  labs(
    x = "",
    y = "unit"
  ) +
  scale_fill_manual(values = c("measure 1" = "lightskyblue4",
                               "measure 2" = "brown",
                               "measure 3" = "goldenrod2",
                               "measure 4" = "turquoise4")) +
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),
    axis.text.y = element_text(size = 12),
    axis.title.y = element_text(size = 25, face = "bold"),
    strip.text = element_text(size = 14, face = "bold")
  ) +
  facet_wrap(~`impact category`, scales = "free") +
  theme(legend.position = "below")
```



```r
# Save plot
ggsave(paste0(location_upload, "absolute_all.png"), width = 20, height = 16,
dpi = 600)
```

Plot all normalized impact categories

```r
# Data preparation
data1 <- results1 %>%
  rename("measure 1" = "norm. SP total") %>%
  select("impact category", "unit", "measure 1")
data2 <- results2 %>%
  rename("measure 2" = "norm. SP FR") %>%
  select("impact category", "measure 2")
data3 <- results3 %>%
  rename("measure 3" = "norm. SP RF") %>%
  select("impact category", "measure 3")
data4 <- results4 %>%
  rename("measure 4" = "norm. SP MS") %>%
  select("impact category", "measure 4")


data <- left_join(data1, data2, by = c("impact category" = "impact category")
)
data <- left_join(data, data3, by = c("impact category" = "impact category"))
data <- left_join(data, data4, by = c("impact category" = "impact category"))

# Save table of all normalized results:
data_table <- data %>%
  mutate(across(
    .cols = c(`measure 1`, `measure 2`, `measure 3`, `measure 4`),
    .fns = ~ format(.x, scientific = TRUE, digits = 3)))
ft <- flextable(data_table) %>%
  theme_vanilla() %>%
  autofit() %>%
  bold(part = "header") %>%
  border_outer() %>%
  border_inner_h() %>%
  border_inner_v()
# Save as PNG
tf <- tempfile(fileext = ".png")
save_as_image(x = ft, path = paste0(location_upload, "_all_results_norm", ".p
ng"))
```

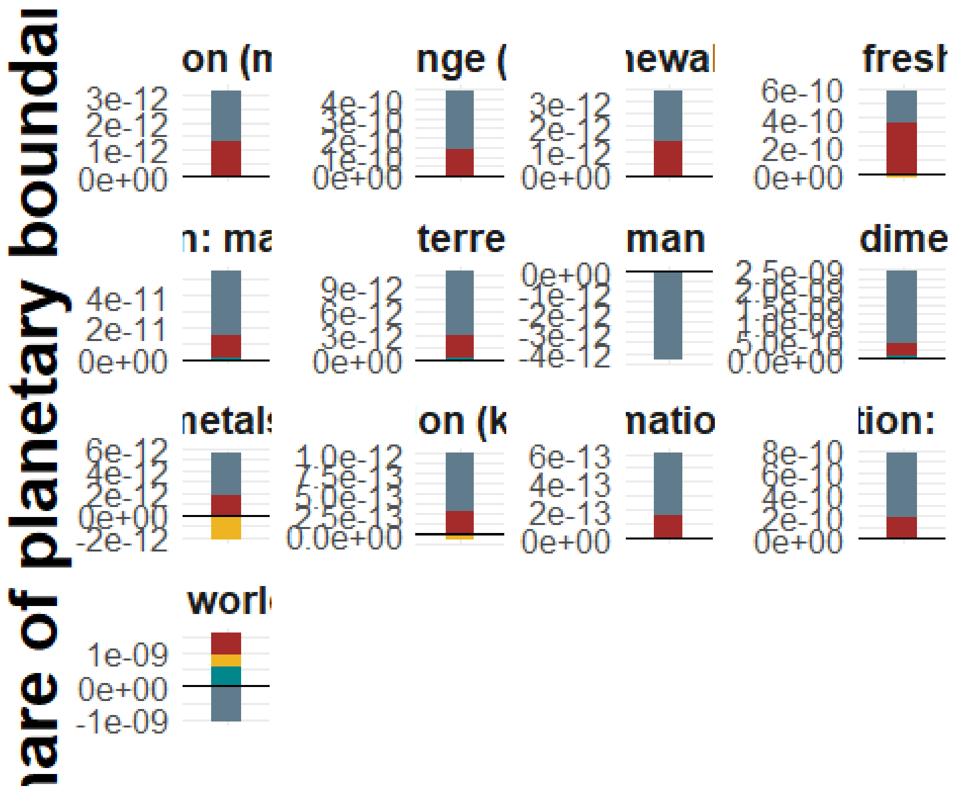
```
## [1] "C:/Users/Klene/Documents/Uni_Bremen/WS24_25/Masterarbeit/R/Results/_a
ll_results_norm.png"
```

```r
# Save table of all normalized results:
data_table <- data %>%
  filter(`impact category` %in% c("Water use", "Land use", "Eutrophication: f
reshwater", "Photochemical oxidant formation: human health", "Climate change"
)) %>%
  mutate(`impact category` = factor(`impact category`, levels = c(
    "Water use",
    "Land use",
    "Eutrophication: freshwater",
    "Photochemical oxidant formation: human health",
    "Climate change"
  ))) %>%
  arrange(`impact category`) %>%  # sortiert nach dem Faktorlevel
  mutate(across(
    .cols = c(`measure 1`, `measure 2`, `measure 3`, `measure 4`),
    .fns = ~ format(.x, scientific = TRUE, digits = 3)))

ft <- flextable(data_table) %>%
  theme_vanilla() %>%
  autofit() %>%
  bold(part = "header") %>%
  border_outer() %>%
  border_inner_h() %>%
  border_inner_v()
# Save as PNG
tf <- tempfile(fileext = ".png")
save_as_image(x = ft, path = paste0(location_upload, "_all_results_norm_relev
ant", ".png"))
```

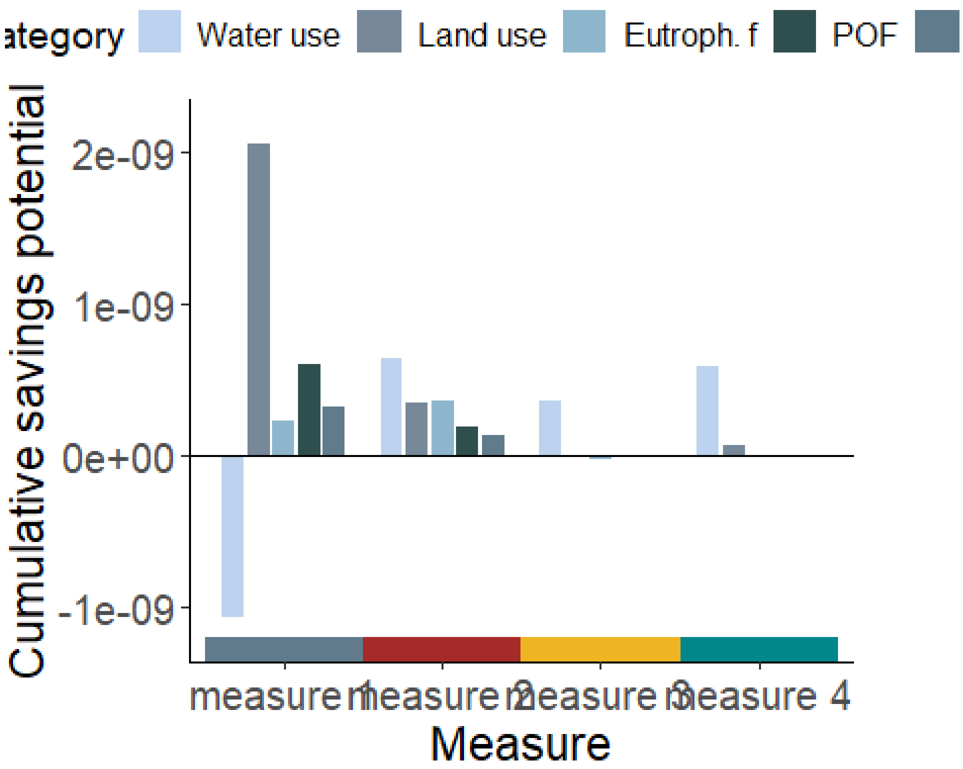
```
## [1] "C:/Users/Klene/Documents/Uni_Bremen/WS24_25/Masterarbeit/R/Results/_a
ll_results_norm_relevant.png"
```

```
write.csv(data_table, paste0(location_upload, "_relevant_norm_data.csv"), row
.names = FALSE)

data <- data %>%
  mutate(`impact category` = paste0(`impact category`, " (", unit, ")"))
data <- data %>%
  select(-unit)


data_long <- data
# data_long <- data_long %>% filter(`impact category` != "Land use" & `impact
category` != "Eutrophication: freshwater")


# data_long$`impact category` <- recode(data_long$`impact category`,
#   "Acidification" = "A",
#   "Climate change" = "CC",
#   "Energy resources: non-renewable" = "nrER",
#   "Eutrophication: freshwater" = "fwE",
#   "Eutrophication: marine" = "mE",
#   "Eutrophication: terrestrial" = "tE",
#   "Ionising radiation: human health" = "IR-hh",
#   "Land use" = "Land use",
#   "Material resources: metals/minerals" = "MR",
#   "Ozone depletion" = "OD",
#   "Particulate matter formation" = "PMF",
#   "Photochemical oxidant formation: human health" = "POF-hh",
#   "Water use" = "Water use"
# )
```

```r
data_long$`impact category` <- recode(data_long$`impact category`,
    "Photochemical oxidant formation: human health" = "Photochemical oxidant
formation: hh")

# Transform data into long format
data_long <- data_long %>%
  gather(key = "measure", value = "value", -`impact category`)

# # Sortiere measure innerhalb jeder `impact category`-Kategorie nach Wertgrö
ße (größte zuerst)
# data_long <- data_long %>%
#   group_by(`impact category`) %>%
#   arrange(desc(value), .by_group = TRUE)  # Sortiere von groß nach klein
#
# # Konvertiere measure in einen Faktor mit den neuen Levels (größte zuerst)
# data_long$measure <- factor(data_long$measure, levels = unique(data_long$me
asure))

# data_long <- data_long %>%
#   arrange(abs(value)) %>%  # Sortiere Werte für jede Kategorie
#   mutate(`impact category` = factor(`impact category`, levels = unique(`imp
act category`)))

ggplot(data_long, aes(x = `impact category`, y = value, fill = measure)) +
  geom_bar(position = "stack", stat = "identity", width = 0.4, show.legend =
TRUE) +
  geom_hline(yintercept = 0, color = "black", linetype = "solid") +
  labs(
    x = "",
    y = "Share of planetary boundary"
  ) +
  scale_fill_manual(values = c("measure 1" = "lightskyblue4",
                               "measure 2" = "brown",
                               "measure 3" = "goldenrod2",
                               "measure 4" = "turquoise4")) +
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),
    axis.text.y = element_text(size = 12),
    axis.title.y = element_text(size = 25, face = "bold"),
    strip.text = element_text(size = 14, face = "bold")
  ) +
  facet_wrap(~`impact category`, scales = "free") +
  theme(legend.position = "below")
```



```
# Save plot
ggsave(paste0(location_upload, "norm_all.png"), width = 20, height = 16, dpi
= 600)

# Conversion of data into long format
data_table <- data_table %>%
  select(`impact category`, unit, `measure 1`, `measure 2`, `measure 3`, `mea
sure 4`) %>%
  pivot_longer(
    cols = starts_with("measure"),
    names_to = "Measure",
    values_to = "SavingsPotential"
  ) %>%
  mutate(SavingsPotential = as.numeric(SavingsPotential))

# Order of measures and impact categories
data_table$Measure <- factor(data_table$Measure, levels = c("measure 1", "mea
sure 2", "measure 3", "measure 4"))
# Rename long impact categories
data_table <- data_table %>%
  mutate(`impact category` = case_when(
    `impact category` == "Photochemical oxidant formation: human health" ~ "P
OF",
    `impact category` == "Eutrophication: freshwater" ~ "Eutroph. f",
    TRUE ~ `impact category`
  ))
data_table$`impact category` <- factor(data_table$`impact category`, levels =
c(
  "Water use", "Land use","Eutroph. f", "POF", "Climate change"
))

ggplot(data_table, aes(x = Measure, y = `SavingsPotential`, fill = `impact ca
tegory`)) +
  annotate("rect", xmin = 0.5, xmax = 1.5, ymin = -Inf, ymax = -1.2e-09, fill
= "lightskyblue4", alpha = 1) +
  annotate("rect", xmin = 1.5, xmax = 2.5, ymin = -Inf, ymax = -1.2e-09, fill
= "brown", alpha = 1) +
  annotate("rect", xmin = 2.5, xmax = 3.5, ymin = -Inf, ymax = -1.2e-09, fill
= "goldenrod2", alpha = 1) +
  annotate("rect", xmin = 3.5, xmax = 4.5, ymin = -Inf, ymax = -1.2e-09, fill
= "turquoise4", alpha = 1) +

  geom_bar(stat = "identity", position = position_dodge(width = 0.8), width =
0.7) +
  geom_hline(yintercept = 0, color = "black", linetype = "solid") +
  scale_fill_manual(values = c("Water use" = "lightsteelblue2",
                              "Land use" = "lightslategrey",
                              "Eutroph. f" = "lightskyblue3",
                              "POF" = "darkslategrey",
                              "Climate change" = "lightskyblue4")) +



                              #c("Water use" = "grey30",
                              # "Land use" = "grey46",
                              # "Eutroph. f" = "grey",
                              # "POF" = "grey79",
                              # "Climate change" = "darkgrey")

                          #c("measure 1" = "lightskyblue4",
                           #    "measure 2" = "brown",
                            #   "measure 3" = "goldenrod2",
                             #  "measure 4" = "turquoise4")
  labs(
    x = "Measure",
    y = "Cumulative savings potential",
    fill = "Impact category"
  ) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 15),
    axis.text.y = element_text(size = 15),
    axis.title.x = element_text(size = 17),
    axis.title.y = element_text(size = 17),
    legend.position = "top",
    legend.title = element_text(size = 14),
    legend.text = element_text(size = 12),
    plot.margin = margin(t = 0, r = 45, b = 0, l = 0)
  ) +
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.09)))
```



```
# Save as PNG
ggsave(paste0(location_upload, "all_relevant_SPs_complex.png"), dpi = 600)

## Saving 5 x 4 in image
```

Upload files

```
# List of Excel files to be uploaded
files_to_upload <- list.files(path = location_upload, pattern = "\\.xlsx$", f
ull.names = TRUE)
# Loop over all files and upload
for (file_path in files_to_upload) {
  # Creating the file name
  file_name <- basename(file_path)
  # Upload file
  response <- PUT(
    url = paste0(nextcloud_url_2, file_name),
    authenticate(username, password),
    body = upload_file(file_path)
  )
  # Check upload status
  if (status_code(response) == 201) {
    print(paste(file_name, "was successfully uploaded."))
  } else if (status_code(response) == 204) {
    print(paste(file_name, "was successfully replaced (no content returned)."
))
```

```
  } else {
    print(paste("Error uploading", file_name, ". Status-Code:", status_code(r
esponse)))
  }
}

# List of PNG files to be uploaded
files_to_upload <- list.files(path = location_upload, pattern = "\\.png$", fu
ll.names = TRUE)
# Loop over all files and upload
for (file_path in files_to_upload) {
  # Create the file name
  file_name <- basename(file_path)
  # Upload file
  response <- PUT(
    url = paste0(nextcloud_url_2, file_name),
    authenticate(username, password),
    body = upload_file(file_path)
  )
  # Check upload status
  if (status_code(response) == 201) {
    print(paste(file_name, "was successfully uploaded."))
  } else if (status_code(response) == 204) {
    print(paste(file_name, "was successfully replaced (no content returned)."
))
  } else {
    print(paste("Error uploading", file_name, ". Status-Code:", status_code(r
esponse)))
  }
}

## [1] "_all_results.png was successfully replaced (no content returned)."
## [1] "_all_results_norm.png was successfully replaced (no content returned)
."
## [1] "_all_results_norm_relevant.png was successfully replaced (no content 
returned)."
## [1] "01_relevant_SP_norm.png was successfully replaced (no content returne
d)."
## [1] "01_relevant_SP_norm_allin1.png was successfully replaced (no content 
returned)."
## [1] "01_relevant_SP_prop.png was successfully replaced (no content returne
d)."
## [1] "01_relevant_SPs_complex.png was successfully replaced (no content ret
urned)."
## [1] "01_results.png was successfully replaced (no content returned)."
## [1] "01_results_norm.png was successfully replaced (no content returned)."
## [1] "01_results_prop.png was successfully replaced (no content returned)."
## [1] "01_results_SPs.png was successfully replaced (no content returned)."
## [1] "02_relevant_SP_norm.png was successfully replaced (no content returne
d)."
## [1] "02_relevant_SP_norm_allin1.png was successfully replaced (no content 
returned)."
## [1] "02_relevant_SP_prop.png was successfully replaced (no content returne
d)."
## [1] "02_results.png was successfully replaced (no content returned)."
## [1] "02_results_norm.png was successfully replaced (no content returned)."
## [1] "02_results_prop.png was successfully replaced (no content returned)."
## [1] "02_results_SPs.png was successfully replaced (no content returned)."
## [1] "03_relevant_SP_norm.png was successfully replaced (no content returne
d)."
## [1] "03_relevant_SP_norm_allin1.png was successfully replaced (no content
returned)."
## [1] "03_relevant_SP_prop.png was successfully replaced (no content returne
d)."
## [1] "03_results.png was successfully replaced (no content returned)."
## [1] "03_results_norm.png was successfully replaced (no content returned)."
## [1] "03_results_prop.png was successfully replaced (no content returned)."
## [1] "03_results_SPs.png was successfully replaced (no content returned)."
## [1] "04_relevant_SP_norm.png was successfully replaced (no content returne
d)."
## [1] "04_relevant_SP_norm_allin1.png was successfully replaced (no content
returned)."
## [1] "04_relevant_SP_prop.png was successfully replaced (no content returne
d)."
## [1] "04_results.png was successfully replaced (no content returned)."
## [1] "04_results_norm.png was successfully replaced (no content returned)."
## [1] "04_results_prop.png was successfully replaced (no content returned)."
## [1] "04_results_SPs.png was successfully replaced (no content returned)."
## [1] "absolute_all.png was successfully replaced (no content returned)."
## [1] "all_relevant_SPs_complex.png was successfully replaced (no content re
turned)."
## [1] "Comparison_CO2.png was successfully replaced (no content returned)."
## [1] "Comparison_norm_CO2.png was successfully replaced (no content returne
d)."
## [1] "norm_all.png was successfully replaced (no content returned)."
```