

1. Bodensaugspannung in nutzbare Feldkapazität (nFK) umwandeln

Samantha Rubo

2021-12-08

Wasserhaltevermögen der Felder der HGU:

Schicht Nr.	Bezeichnung	100% nFK entsprechen	“Wassergehalt (Vol.% bei 100% nFK)”
1	0-30 cm	49.3 mm	16.4%
2	30-60 cm	46.1 mm	15.4%
3	60-90 cm	43.4 mm	14.5%

Daten unter “Z:\Außenbetrieb\Flächenbelegung\Wasserhaltevermögen Böden Felder Gb.xlsx”

```
knitr::include_graphics("../graphics/Feldkapazität_Thomas.png")
```

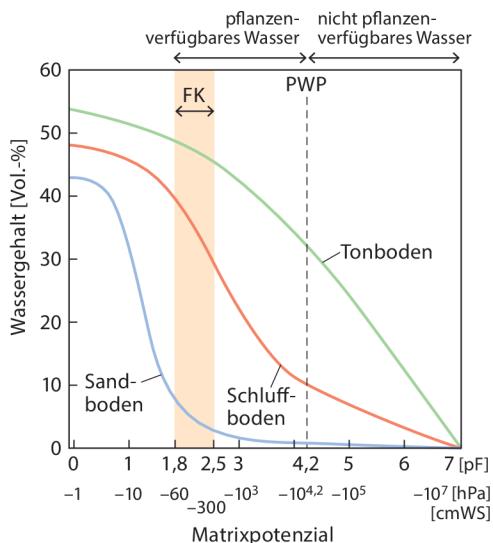


Figure 1: “Beziehung Volumetrischer Wassergehalt ~ Matrixpotential. Berechnungsgrundlage der nFK. Quelle: Thomas F. (2018) Ökophysiologische Leistungen der Höheren Pflanzen. In: Grundzüge der Pflanzenökologie. Springer Spektrum, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-54139-5_3”

Grenzwerte zur Berechnung der nutzbaren Feldkapazität

```
pf_min <- 1.8
pf_max <- 4.2
hPa_min <- 10^pf_min
hPa_max <- 10^pf_max
# Vol% Wasser bei 100% nFK für drei Bodentiefen:
T0020 <- 49.3
```

```
T4060 <- 46.1
T2040 <- T0020 - (T0020 - T4060) / 2
```

Tensiometer-Daten, Wetter und Bewässerung aus SQLite-Datenbank lesen.

```
# Verbindung zur Datenbank herstellen:
path1 <- ifelse(Sys.info()["nodename"] == "Samantha-MBP.local",
                 "../../../Github_GeoSenSys/GeoSenSys2020/", "../../GeoSenSys2020/")
)
db <- paste0(path1, "Data_2020/HGU_GeoSenSys_V3_6.db") # DB in other R-Project

db1 <- dbConnect(RSQLite::SQLite(), db)

# Query fuer Tensiometer-Datensatz
query <- "SELECT
            Spinat_Saetze.satz_id,
            Varianten.variante_acronym,
            Varianten.variante_H20,
            Parzellen.wiederholung,
            Tensiometer.zeit_messung,
            Tensiometer.bodensaugspannung_0020_hPa,
            Tensiometer.bodensaugspannung_2040_hPa,
            Tensiometer.bodensaugspannung_4060_hPa

            FROM Tensiometer
            LEFT JOIN Parzellen ON Tensiometer.parzelle_id = Parzellen.parzelle_id
            LEFT JOIN Varianten ON Parzellen.variante_id = Varianten.variante_id
            LEFT JOIN Spinat_Saetze ON Varianten.satz_id = Spinat_Saetze.satz_id
            WHERE Varianten.variante_N = 'N100'
            ;" #zunächst nur fuer Stickstoff-vollversorgte Varianten
tensio <- dbGetQuery(db1, query)

#### Query fuer Wetter-Daten:
query2 <- "SELECT
            Wetter.satz_id,
            Wetter.datum_wetter,
            Wetter.niederschlag_mm
            FROM
            Wetter"

wetter <- dbGetQuery(db1, query2) %>% # Niederschlag aller Saetze einlesen
  mutate_at("datum_wetter", ~ as_date(.))

# Query fuer Bewässerungs-Datensatz
query3 <- "SELECT
            Spinat_Saetze.satz_id,
            Varianten.variante_acronym,
            Varianten.variante_H20,
            Parzellen.wiederholung,
            Bewässerung.datum_bewässerung,
            Bewässerung.wassermengen_mm
```

```

    FROM Bewaesserung
    LEFT JOIN Parzellen ON Bewaesserung.porzelle_id = Parzellen.porzelle_id
    LEFT JOIN Varianten ON Parzellen.variante_id = Varianten.variante_id
    LEFT JOIN Spinat_Saetze ON Varianten.satz_id = Spinat_Saetze.satz_id
    WHERE Varianten.variante_N = 'N100'
    "
bewaesserung <- dbGetQuery(db1, query3) %>% # Bewaesserung aller Saetze einlesen
  mutate_at("datum_bewaesserung", ~ as_date(.))

dbDisconnect(db1) # Verbindung zur Datenbank beenden
rm(db, db1, path1, query, query2, query3) # Helfer-Objekte loeschen

```

Daten formatieren und Tagesmittelwerte der Bodensaugspannung berechnen

```

tensio <- tensio %>%
  # Datum formatieren und Tagesmittelwerte bilden
  mutate_at("zeit_messung", ~ as_datetime(.) %>%
    format.Date(., format = "%Y-%m-%d") %>% # fuer Tages-Mittelwert
    as_date(.)) %>% # wieder in Datum (class) umformen
  # Faktorstufen sortieren (für Grafik)
  mutate_at("variante_H20", ~ factor(., levels = c("Wfull_plus", "Wfull", "Wred"))) %>%
  group_by(satz_id, variante_H20, wiederholung, zeit_messung) %>%
  summarise_at(c(
    "bodensaugspannung_0020_hPa",
    "bodensaugspannung_2040_hPa",
    "bodensaugspannung_4060_hPa"
  ), ~ round(mean(., na.rm = TRUE), digits = 2))

```

hPa in pf umwandeln: $\log_{10}(\text{hPa})$

```

tensio <- tensio %>%
  mutate(
    pf_0020 = log10(bodensaugspannung_0020_hPa),
    pf_2040 = log10(bodensaugspannung_2040_hPa),
    pf_4060 = log10(bodensaugspannung_4060_hPa)
  )

```

nFK berechnen

```

nfk_fun <- function(x) {
  (1 - (x - pf_min) / (pf_max - pf_min)) * 100
}
tensio <- tensio %>% # as_tibble() %>%
  mutate_at(
    c("pf_0020", "pf_2040", "pf_4060"),
    list(nFK = nfk_fun)
  ) %>%
  rename_with(.cols = ends_with("_nFK"), ~ gsub("pf_", "T", .x, fixed = TRUE)) %>%
  # Tabelle komprimieren: 2 Nachkommastellen
  mutate_if(is.numeric, ~ round(., 2))

```

Tabelle formatieren für ggplot

```
tensio_melted <- tidyr::pivot_longer(tensio %>%
  select(c(
    "satz_id", "variante_H20", "wiederholung", "zeit_messung",
    "T0020_nFK", "T2040_nFK", "T4060_nFK"
  )),
  cols = c("T0020_nFK", "T2040_nFK", "T4060_nFK"),
  names_to = "BodenTiefe",
  values_to = "nFK_prozent"
) %>%
  mutate(BodenTiefe = substr(BodenTiefe, 4, 5) %>% as.numeric()) %>%
  mutate(kategorie = factor("nFK", levels = c("wasserinput", "nFK")))

#Tensio bei 0cm erweitern
tensio0 <- tensio_melted %>%
  filter(BodenTiefe == 20) %>%
  mutate(BodenTiefe = 0)

tensio_melted <- bind_rows(tensio_melted, tensio0)

rm(tensio0)
```

Wetter- und Bewässerungsdaten formatieren und zusammenführen

```
#Wenn nach Wiederholung unterschieden, dann kein Mittelwert. Wenn nach Variante, dann Mittelwert
bewaesserung_mean <- bewaesserung %>% rename(bewaesserung_mm = wassermengen_mm)

# Bewässerung und Niederschlag in eine Tabelle zusammenfuehren
wasser_gesamt <- tensio %>%
  select(satz_id, variante_H20, wiederholung, zeit_messung) %>%
  left_join(wetter, by = c("satz_id", "zeit_messung" = "datum_wetter")) %>%
  left_join(bewaesserung_mean, by = c("satz_id", "variante_H20", "wiederholung",
    "zeit_messung" = "datum_bewaesserung"))
)) %>%
# Faktorstufen sortieren (für Grafik)
tidyr::pivot_longer(
  cols = c("bewaesserung_mm", "niederschlag_mm"),
  names_to = "variable", values_to = "value"
) %>%
  mutate(kategorie = factor("wasserinput", levels = c("wasserinput", "nFK")) %>%
  mutate_at("variante_H20", ~ factor(., levels = c("Wfull_plus", "Wfull", "Wred"))))

#stacked values anfuegen fuer plot:
stacked_bars_plot <- function(data, gruppen){
  data %>%
    group_by_at(vars(gruppen)) %>%
    mutate_at("value", ~ifelse(is.na(.), 0, .)) %>%
    mutate(value_stacked = cumsum(value)) %>%
    mutate(value_min = ifelse(variable == "bewaesserung_mm", 0, value[1])) %>%
    mutate_at(c("value_stacked", "value_min"), ~ifelse( value == 0, NA, .)) %>%
```

```

ungroup()
}

wasser_gesamt <- stacked_bars_plot(data = wasser_gesamt,
                                     gruppen = c("kategorie", "satz_id", "variante_H20",
                                                "wiederholung", "zeit_messung"))

#Standardabweichung der Wiederholung ###Daten
#filter: satz_id == satz_nr, variante_H20 == variante
satz_nr = 5; variante = "Wfull_plus"
tensio_sd <- tensio_melted %>%
  pivot_wider(id_cols = c("kategorie",
                         "satz_id", "variante_H20", ##
                         "zeit_messung", "Bodentiefe"),
              names_from = "wiederholung", values_from = "nFK_prozent") %>%
  rowwise() %>%
  mutate(nFK_sd = sd(c(a,b,c,d), na.rm = TRUE)) %>%
  select(-a, -b, -c, -d) %>%
  ungroup

wasser_gesamt_sd <- wasser_gesamt %>%
  filter(variable != "niederschlag_mm") %>%
  #niederschlag_mm ausschließen, da alle Wiederholungen gleich.
  pivot_wider(id_cols = c("kategorie",
                         "satz_id", "variante_H20", ##
                         "variable", "zeit_messung"),
              names_from = "wiederholung", values_from = "value") %>%
  rowwise() %>%
  mutate(wasser_sd = sd(c(a,b,c,d), na.rm = TRUE)) %>%
  mutate_at("wasser_sd", ~ifelse(.==0, NA,.)) %>%
  select(-a, -b, -c, -d) %>%
  ungroup

```

nFK plotten

```

# plot erstellen: #minus-Bodentiefe, da der Wert die "bis"-Bodentiefe beschreibt
plot_nfk <- function(satz_nr, subtitle, wdh = FALSE, variante = NULL) {
  #plot_nfk(satz_nr = 5, subtitle = "2021, Satz 3, Feld 6", wdh = TRUE, variante = "Wfull_plus")

  #Pro Variante oder einzeln fuer Wiederholung:
  if(isFALSE(wdh)) { #== Mittelwert aller Wiederholungen einer Variante

    tensio_melted <- tensio_melted %>%
      filter(satz_id == satz_nr) %>%
      group_by(kategorie, satz_id, variante_H20, Bodentiefe, zeit_messung) %>%
      summarise(nFK_prozent = mean(nFK_prozent, na.rm = TRUE)) %>%
      ungroup();

    wasser_gesamt <- wasser_gesamt %>%
      filter(satz_id == satz_nr) %>%
      group_by(kategorie, variable, satz_id, variante_H20, zeit_messung) %>%
      summarise(value = mean(value, na.rm = TRUE)) %>%

```

```

ungroup() %>%
stacked_bars_plot(gruppen = c("kategorie", "satz_id", "variante_H2O","zeit_messung"));

} else { #== Wiederholungen einer Variante vergleichen
tensio_melted <- tensio_melted %>% filter(satz_id == satz_nr & variante_H2O == variante);
wasser_gesamt <- wasser_gesamt %>% filter(satz_id == satz_nr & variante_H2O == variante);

#SD-Daten
wasser_gesamt_sd2 <- wasser_gesamt_sd %>%
  filter(satz_id == satz_nr & variante_H2O == variante) %>%
  mutate(wiederholung = "Stabw")

tensio_sd2 <- tensio_sd %>%
  filter(satz_id == satz_nr & variante_H2O == variante) %>%
  mutate(wiederholung = "Stabw")

#Farben des SD-Plots (Grau-Skala)
my_breaks_sd <- c(seq(from = 0, to = 30, by = 5), Inf);
my_colors_sd <- grey.colors(7, start = 0.1, end = 0.8, rev = TRUE);
my_labels_sd <- levels(cut(tensio_sd2$nfK_sd, breaks = my_breaks_sd))
}

#Farbspektrum definieren für nFK:
my_breaks <- c(seq(from = 0, to = 120, by = 10), Inf)
my_colors <- c(
  colorRampPalette(c("brown", "#fac83c"))(6),
  colorRampPalette(c("lightgreen", "forestgreen"))(3),
  colorRampPalette(c("#32c8fa", "darkblue"))(4)
)
my_labels <- levels(cut(tensio_melted$nFK_prozent, breaks = my_breaks))

#gg-Plot-Funktion:
p <- ggplot() +
  # nFK anfügen:
  geom_contour_filled(
    data = tensio_melted,
    aes(x = zeit_messung, y = -(Bodentiefe), z = nFK_prozent),
    breaks = my_breaks,
  ) +
  scale_fill_manual("Bodenfeuchte (% nFK)",
                    values = my_colors,
                    labels = my_labels,
                    drop = FALSE
  ) +
  # "drop = FALSE" ist notwendig um auch Werte anzuzeigen (und die Farben zu scalen),
  # die nicht ueber den Wertebereich in der CSV hinausgehen (z.B. <10 oder >120 % nFK)
  labs(title = "Bodenfeuchte und Wassereintrag Spinat-Versuch",
       subtitle = subtitle, x = "Date", y = ""
  ) +

```

```

# Regen und Bewässerung anfügen:
geom_segment(data=wasser_gesamt,
              aes(x = zeit_messung, y = value_min,
                  xend = zeit_messung, yend = value_stacked,
                  position = "dodge",
                  # geom_col(data=wasser_gesamt,
                  #           aes(x = zeit_messung, y= value,
                  #               color = variable), fill=NA, size=1.5) +
scale_color_manual("Wassereintrag (mm)",
                   values = c("niederschlag_mm" = "gray70",
                             "bewaesserung_mm" = "gray10"),
                   labels = c("niederschlag_mm" = "Regen",
                             "bewaesserung_mm" = "GS"))
)

# Standardabweichung anfügen:
p <- if(isTRUE(wdh)){
  p+ ggnewscale::new_scale("fill") + # zweite fill-scale für SD-Plot
    geom_contour_filled(
      data = tensio_sd2,
      aes(x = zeit_messung, y = -(Bodentiefe), z = nfK_sd),
      breaks = my_breaks_sd,
    ) +
    scale_fill_manual("Stabw Bodenfeuchte (% nFK)",
                      values = my_colors_sd,
                      labels = my_labels_sd,
                      drop = FALSE
    ) +
    geom_segment(data=wasser_gesamt_sd2,
                aes(x = zeit_messung, y =0,
                    xend = zeit_messung, yend = wasser_sd,
                    color = variable),fill=NA, size=1)
} else {p}

# Facets anfügen:
p<- p+
  facet_grid(kategorie ~ if(isFALSE(wdh)){variante_H2O} else {wiederholung},
              scales = "free_y", switch = "y", space = "free_y",
              labeller = labeller(
                .rows = as_labeller(
                  c(nFK = "Bodentiefe (cm)",
                    wasserinput = "Wassereintrag (mm)")),
                .cols = label_value),
                drop = FALSE # damit Wfull_plus in Satz1 dargestellt wird (ohne Daten)
  ) +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    strip.placement = "outside",
    strip.background.y = element_blank(),

```

```

    panel.spacing = unit(0, "lines"),
    legend.key.size = unit(12, units = "pt"),
    legend.key.width = unit(12, units = "pt")
) +
guides(col = guide_legend(order = 1))

# Ausgabe des Plots
p
}

#Plot-Funktion ausfuehren
p2 <- plot_nfk(satz_nr = 2, subtitle = "2020, Satz 2, Feld 4a (*Paper)", wdh = FALSE)
p3 <- plot_nfk(satz_nr = 3, subtitle = "2021, Satz 1, Feld 6", wdh = FALSE)
p4 <- plot_nfk(satz_nr = 4, subtitle = "2021, Satz 2, Feld 4b", wdh = FALSE)
p5 <- plot_nfk(satz_nr = 5, subtitle = "2021, Satz 3, Feld 6", wdh = FALSE)

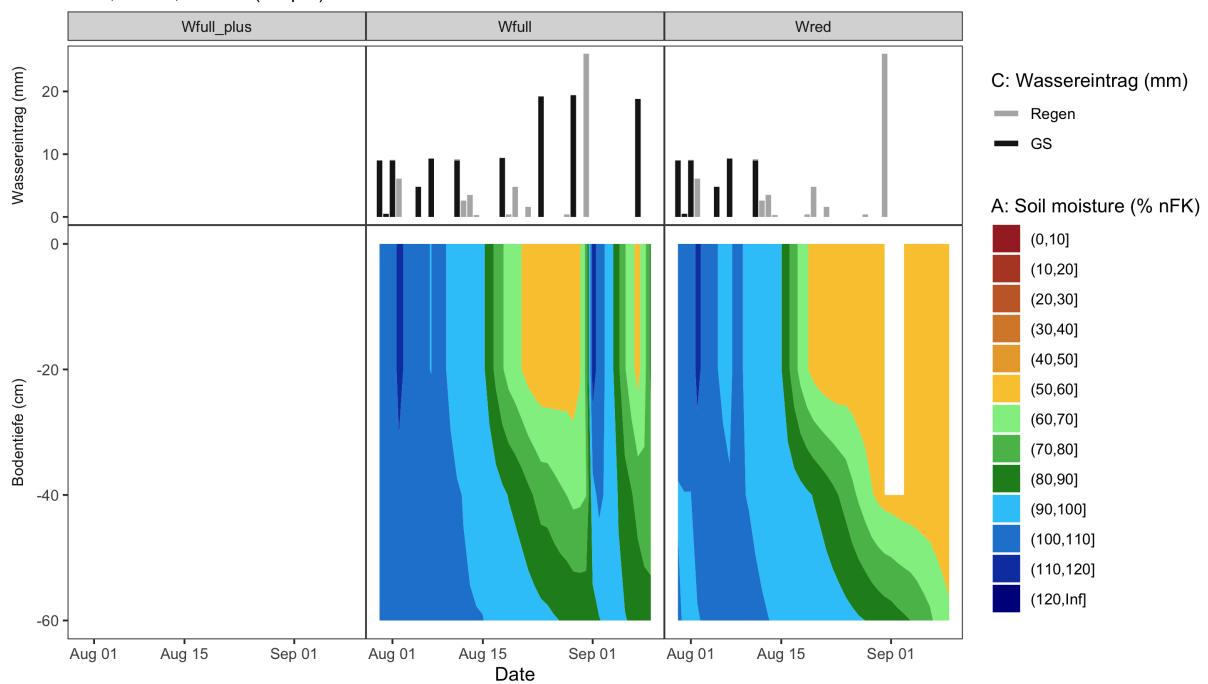
p2; p3; p4; p5

#Grafik speichern
# file_list <- list(
#   #file1 <- #keine Tensiometer-Daten fuer Satz 1
#   file2 = ".../graphics/nFK_2020_Satz2.png",
#   file3 = ".../graphics/nFK_2021_Satz1.png",
#   file4 = ".../graphics/nFK_2021_Satz2.png",
#   file5 = ".../graphics/nFK_2021_Satz3.png")
#
# purrrr::map2(file_list, list(p2, p3, p4, p5),
#   ~ggsave(filename = .x, plot = .y, device = "png", width = 10, height = 6, dpi = 300)
# )

```

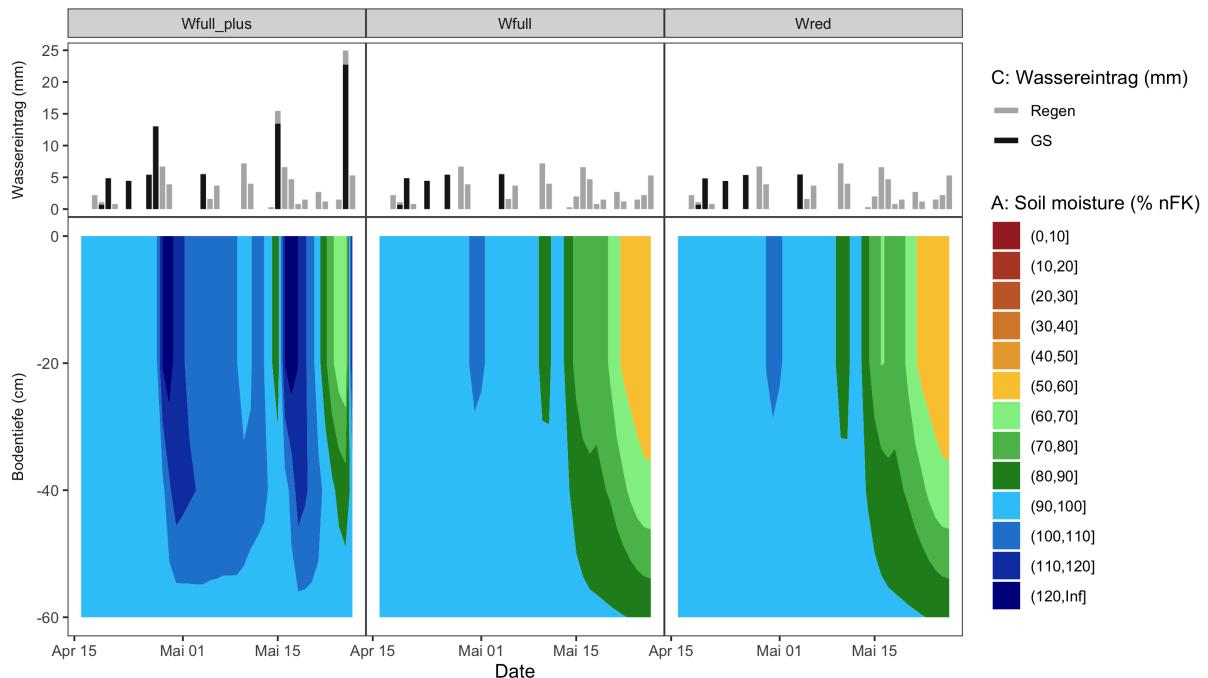
Bodenfeuchte und Wassereintrag Spinat-Versuch

2020, Satz 2, Feld 4a (*Paper)



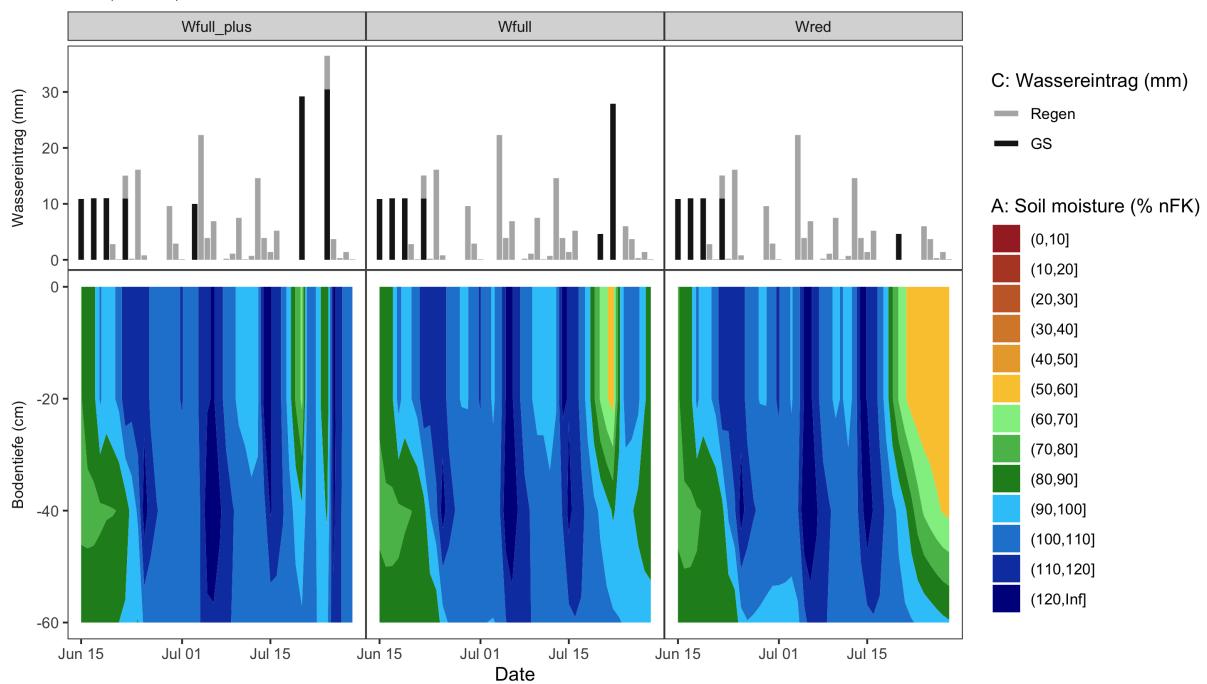
Bodenfeuchte und Wassereintrag Spinat-Versuch

2021, Satz 1, Feld 6



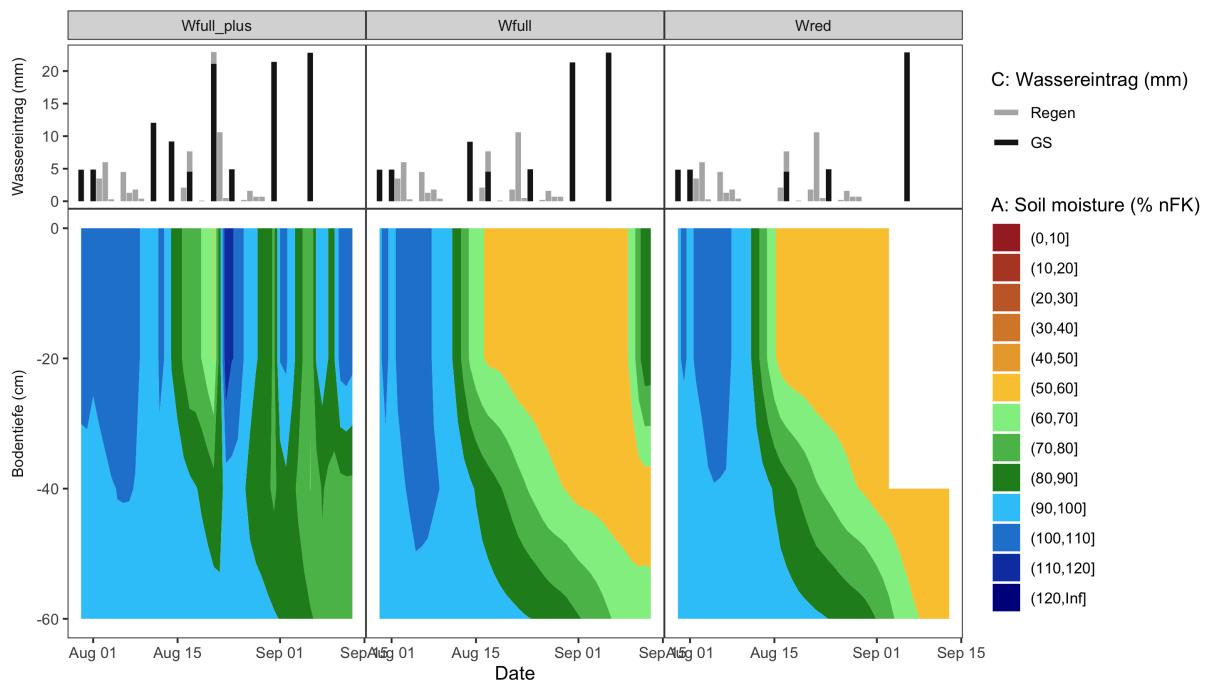
Bodenfeuchte und Wassereintrag Spinat-Versuch

2021, Satz 2, Feld 4b



Bodenfeuchte und Wassereintrag Spinat-Versuch

2021, Satz 3, Feld 6



Plots pro Variante (== Vergleich der Wiederholungen):

#2020 Satz2: nur an zwei Stellen Tensiometer: keine Wiederholung.

```

#2021 Satz1
p1 <- plot_nfk(satz_nr = 3, subtitle = "2021, Satz 1, Feld 6, Variante 'Wfull_plus'", 
                wdh = TRUE, variante = "Wfull_plus")
p2 <- plot_nfk(satz_nr = 3, subtitle = "2021, Satz 1, Feld 6, Variante 'Wfull'", 
                wdh = TRUE, variante = "Wfull")
p3 <- plot_nfk(satz_nr = 3, subtitle = "2021, Satz 1, Feld 6, Variante 'Wred'", 
                wdh = TRUE, variante = "Wred")

#2021 Satz2
p4 <- plot_nfk(satz_nr = 4, subtitle = "2021, Satz 2, Feld 4b, Variante 'Wfull_plus'", 
                wdh = TRUE, variante = "Wfull_plus")
p5 <- plot_nfk(satz_nr = 4, subtitle = "2021, Satz 2, Feld 4b, Variante 'Wfull'", 
                wdh = TRUE, variante = "Wfull")
p6 <- plot_nfk(satz_nr = 4, subtitle = "2021, Satz 2, Feld 4b, Variante 'Wred'", 
                wdh = TRUE, variante = "Wred")

#2021 Satz3
p7 <- plot_nfk(satz_nr = 5, subtitle = "2021, Satz 3, Feld 6, Variante 'Wfull_plus'", 
                wdh = TRUE, variante = "Wfull_plus")
p8 <- plot_nfk(satz_nr = 5, subtitle = "2021, Satz 3, Feld 6, Variante 'Wfull'", 
                wdh = TRUE, variante = "Wfull")
p9 <- plot_nfk(satz_nr = 5, subtitle = "2021, Satz 3, Feld 6, Variante 'Wred'", 
                wdh = TRUE, variante = "Wred")

#Grafiken der Wiederholung speichern
file_list <- list(
  file1 = ".../graphics/nFK_2021_Satz1_Wfull_plus.png",
  file2 = ".../graphics/nFK_2021_Satz1_Wfull.png",
  file3 = ".../graphics/nFK_2021_Satz1_Wred.png",
  file4 = ".../graphics/nFK_2021_Satz2_Wfull_plus.png",
  file5 = ".../graphics/nFK_2021_Satz2_Wfull.png",
  file6 = ".../graphics/nFK_2021_Satz2_Wred.png",
  file7 = ".../graphics/nFK_2021_Satz3_Wfull_plus.png",
  file8 = ".../graphics/nFK_2021_Satz3_Wfull.png",
  file9 = ".../graphics/nFK_2021_Satz3_Wred.png")

purrr::map2(file_list, list(p1,p2,p3,p4,p5,p6,p7,p8,p9),
            ~ggsave(filename = .x, plot = .y, device = "png", width = 12, height = 6, dpi = 300)
)

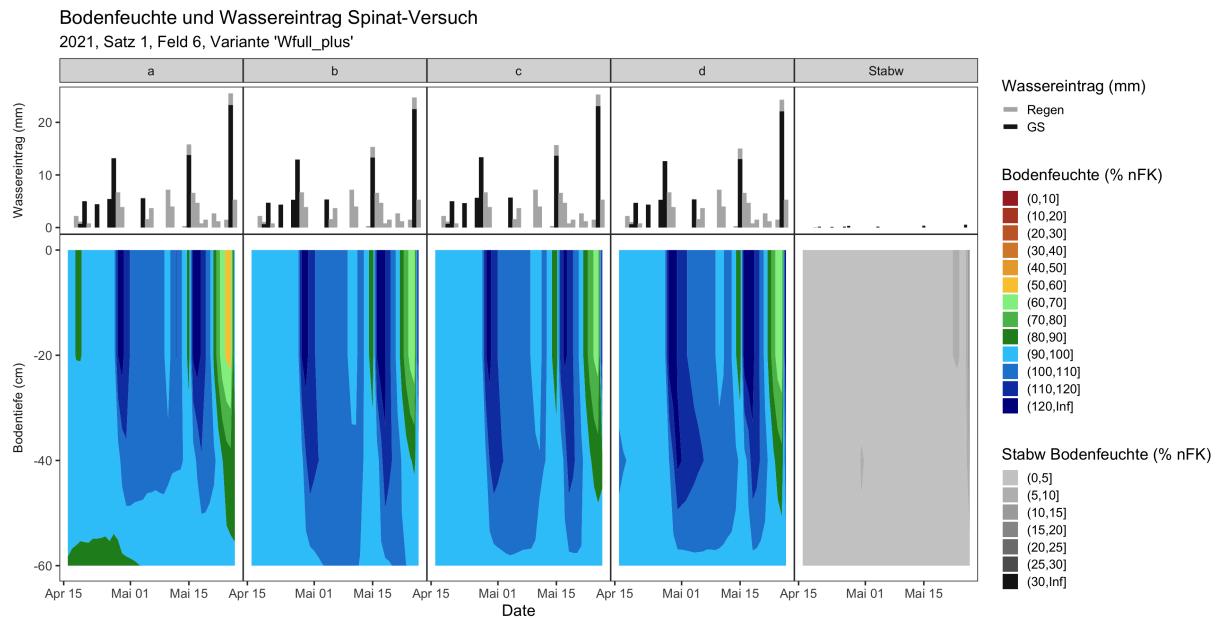
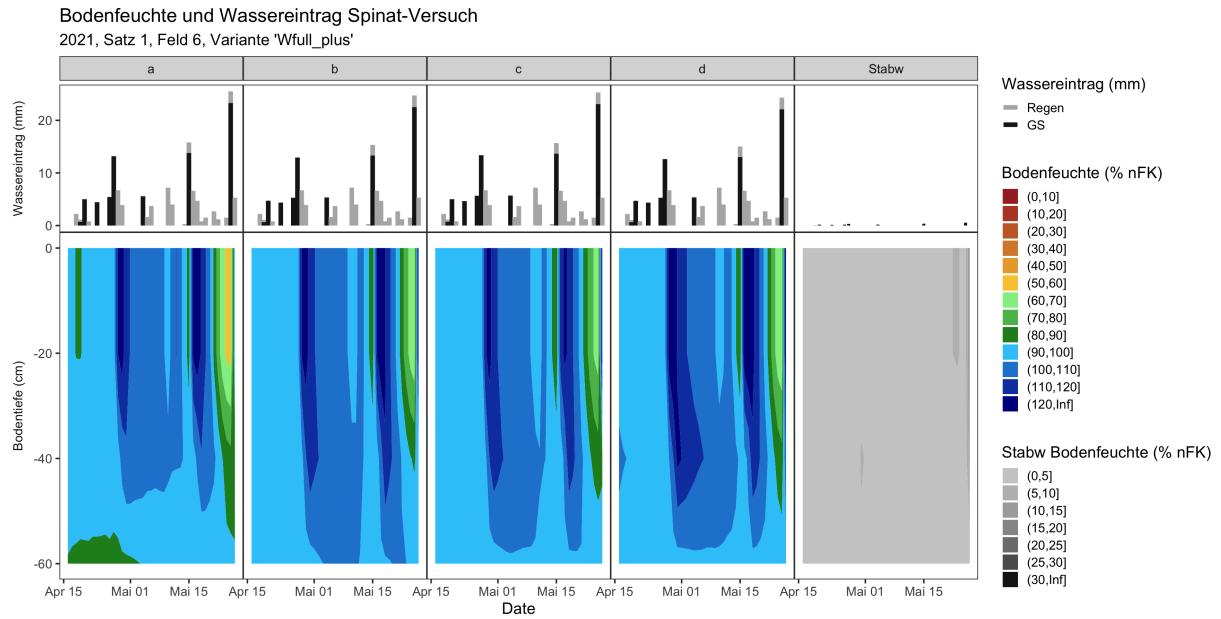
## $file1
## [1] ".../graphics/nFK_2021_Satz1_Wfull_plus.png"
##
## $file2
## [1] ".../graphics/nFK_2021_Satz1_Wfull.png"
##
## $file3
## [1] ".../graphics/nFK_2021_Satz1_Wred.png"
##
## $file4
## [1] ".../graphics/nFK_2021_Satz2_Wfull_plus.png"
##
## $file5
## [1] ".../graphics/nFK_2021_Satz2_Wfull.png"

```

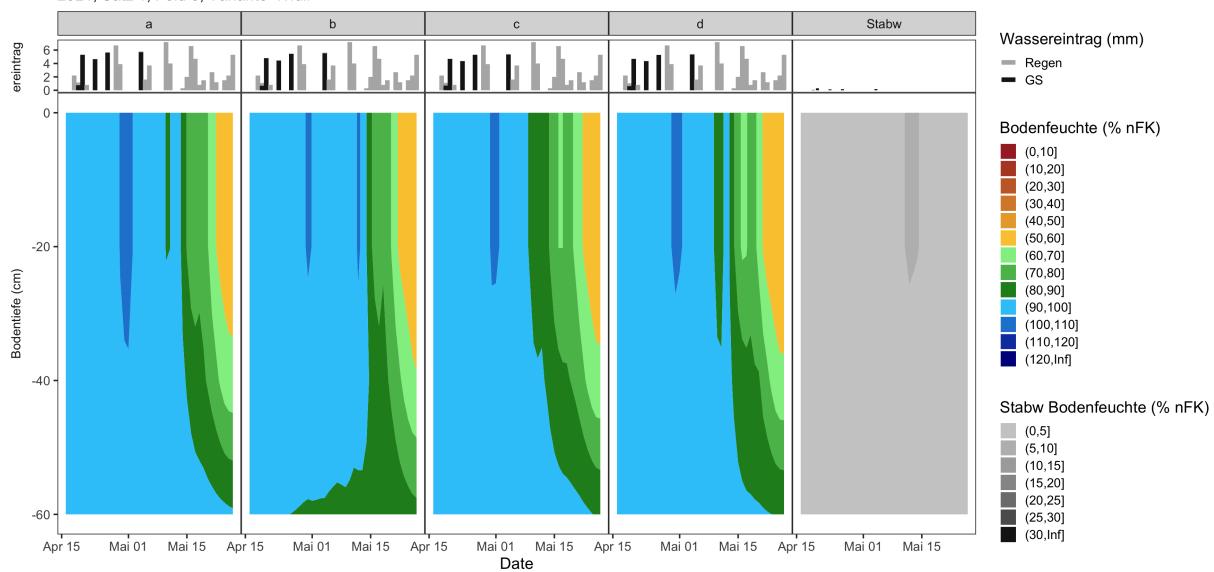
```

## 
## $file6
## [1] "../graphics/nFK_2021_Satz2_Wred.png"
##
## $file7
## [1] "../graphics/nFK_2021_Satz3_Wfull_plus.png"
##
## $file8
## [1] "../graphics/nFK_2021_Satz3_Wfull.png"
##
## $file9
## [1] "../graphics/nFK_2021_Satz3_Wred.png"

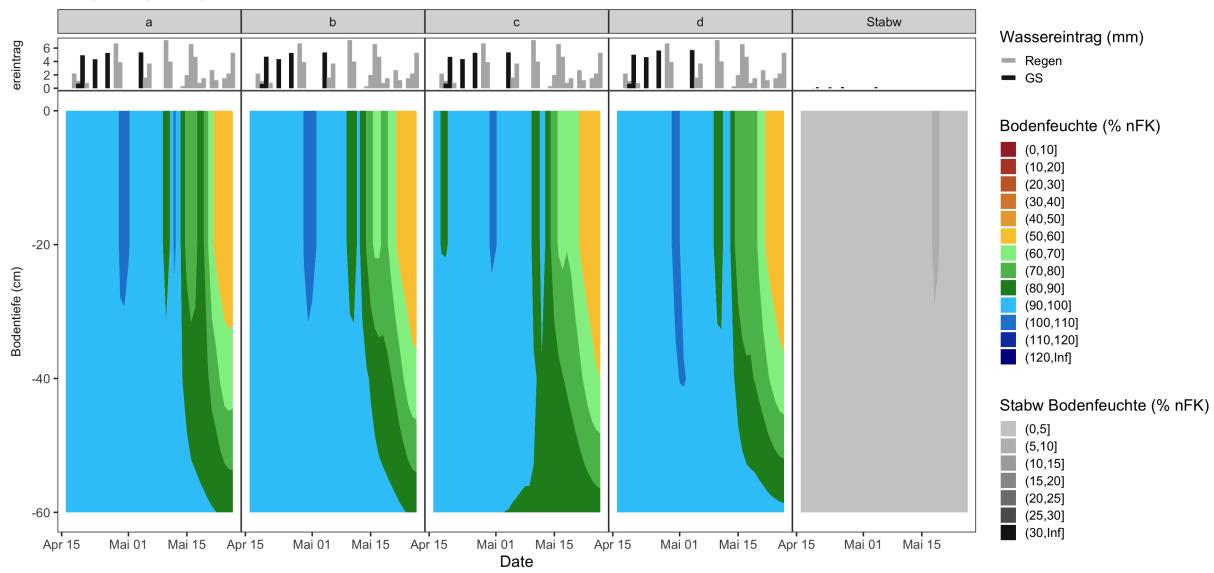
```



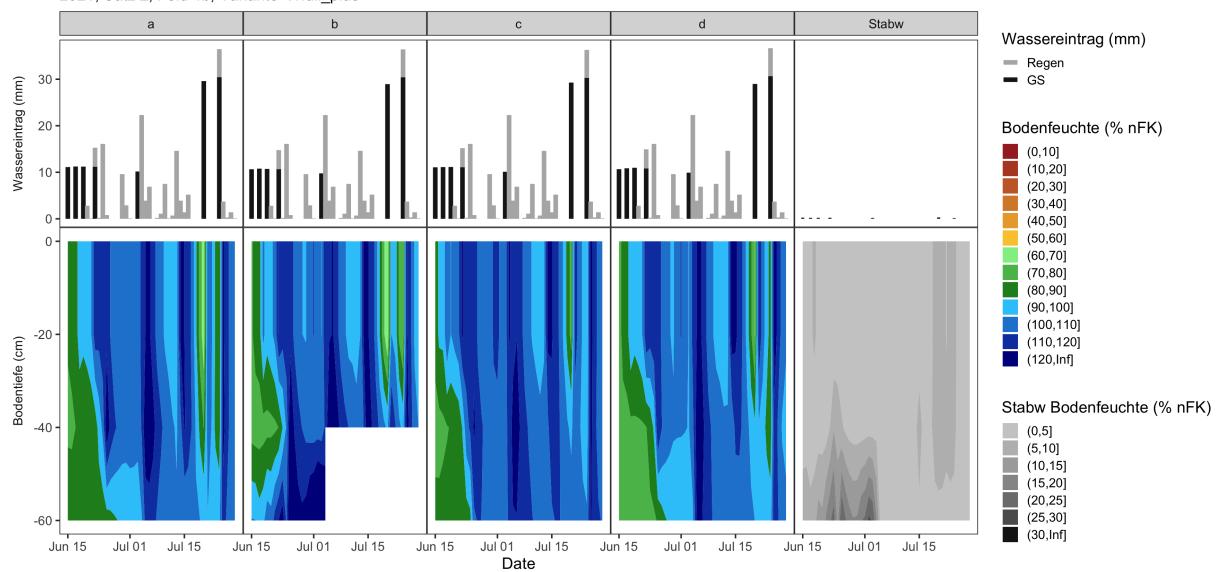
Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 1, Feld 6, Variante 'Wfull'



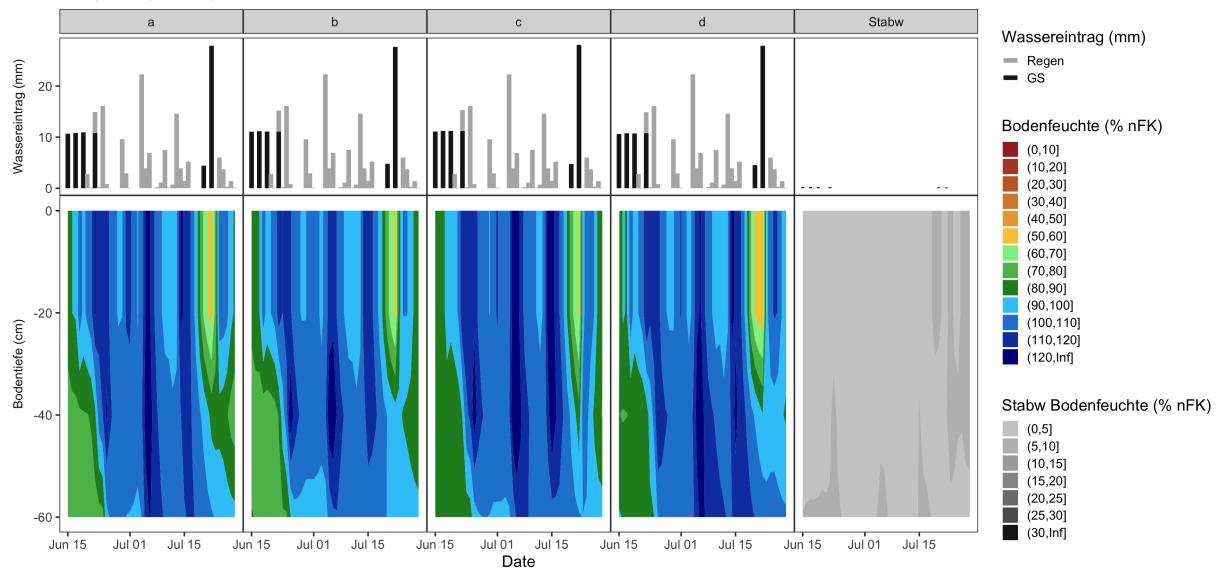
Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 1, Feld 6, Variante 'Wred'



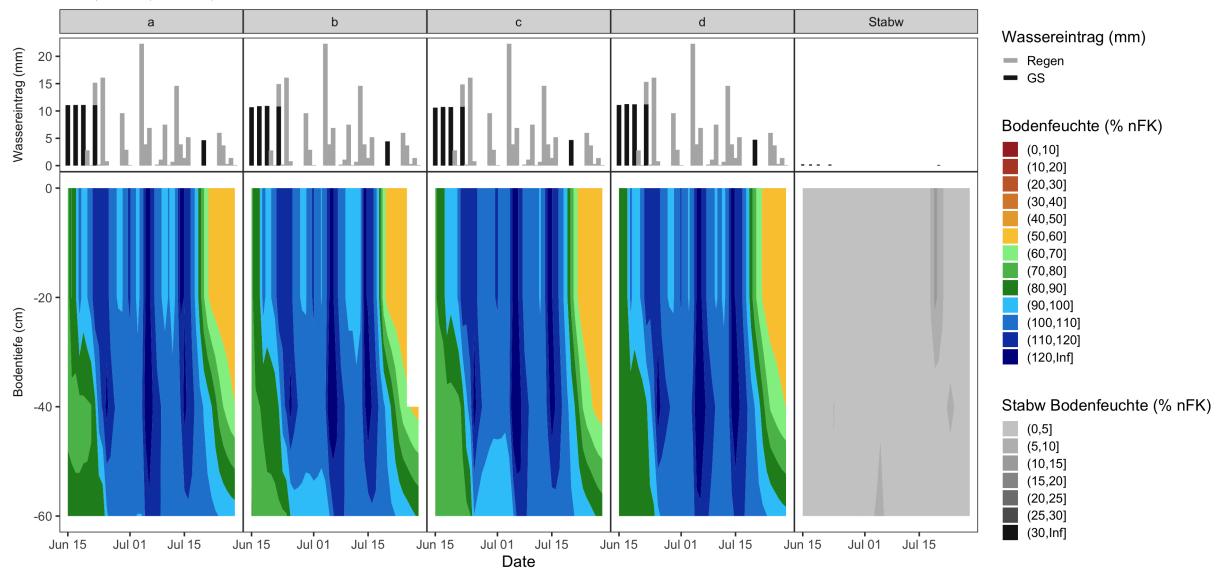
Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 2, Feld 4b, Variante 'Wfull_plus'



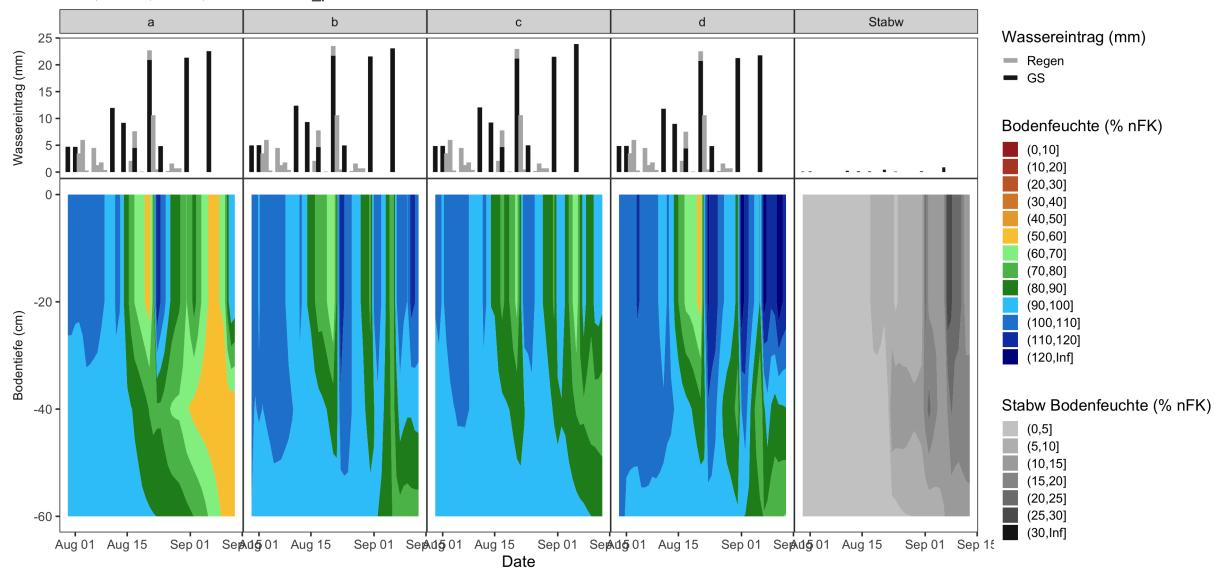
Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 2, Feld 4b, Variante 'Wfull'



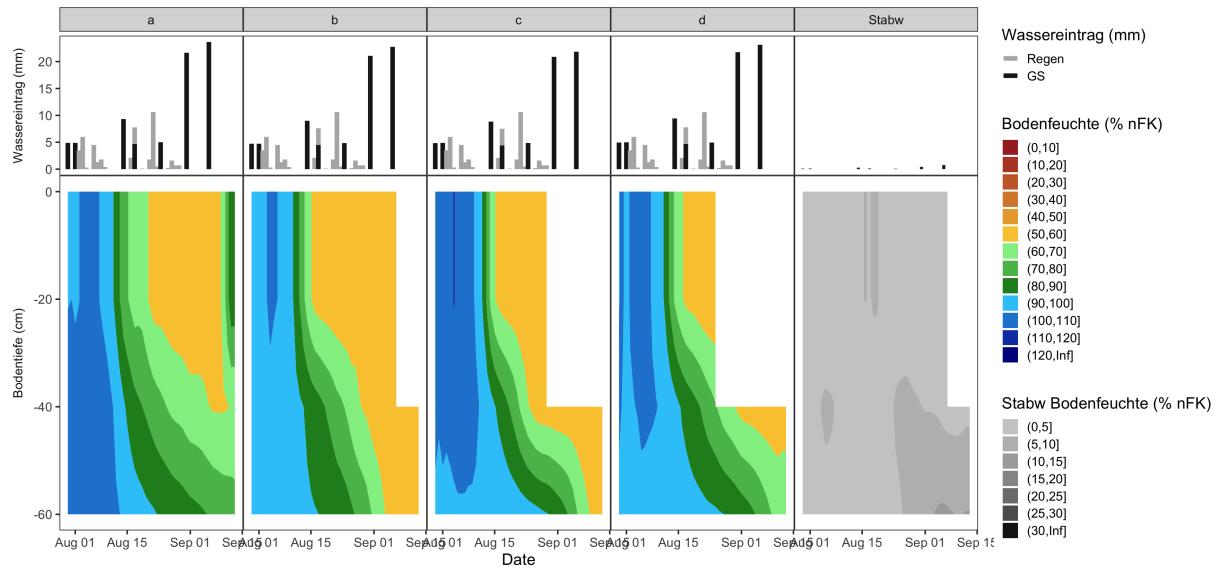
Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 2, Feld 4b, Variante 'Wred'



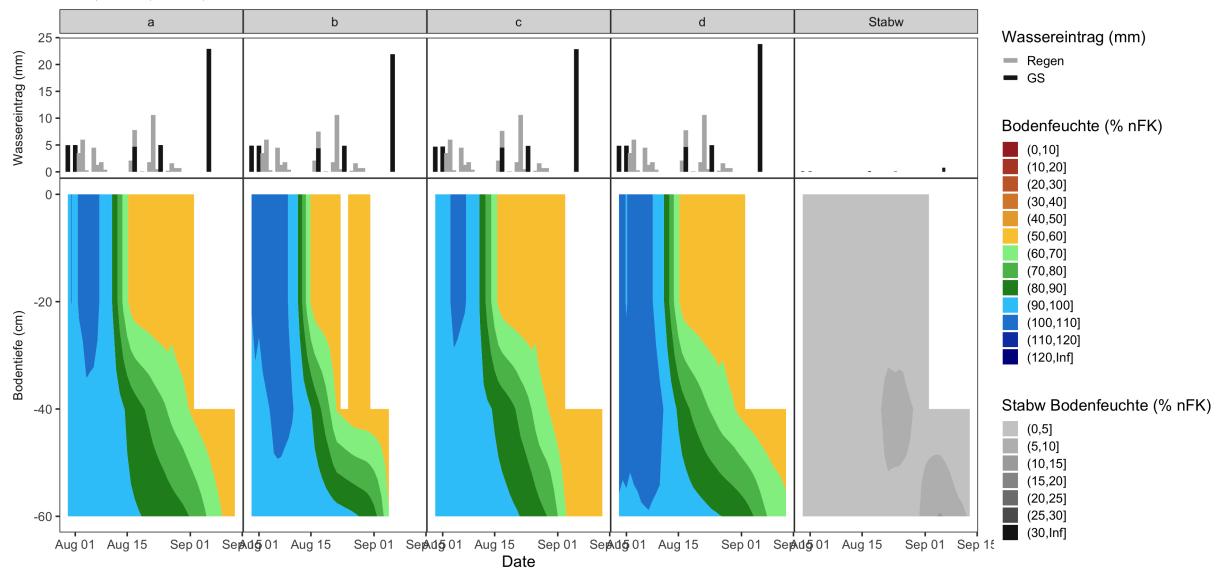
Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 3, Feld 6, Variante 'Wfull_plus'



Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 3, Feld 6, Variante 'Wfull'



Bodenfeuchte und Wassereintrag Spinat-Versuch
2021, Satz 3, Feld 6, Variante 'Wred'



#nFK-Tabelle speichern

```
# file1 <- "../data/derived_data/nfK_2020_2021_20220309.csv"
# data.table::fwrite(x = tensio, file = file1, sep = ";", dec = ".")
```

#Bewässerung_mean-Tabelle speichern

```
# file1 <- "../data/derived_data/bewaesserung_mean_2020_2021_20220309.csv"
# data.table::fwrite(x = bewaesserung_mean, file = file1, sep = ";", dec = ".")
```