

Red Fuji

Tran Hung Dat

2025-05-13

```
# install.packages("tidyverse")
# install.packages("readr")
# install.packages("ggplot2")
# install.packages("grid") # part of base R, no need to install
# install.packages("dplyr")
# install.packages("scales")
# install.packages("colorspace")
# install.packages("magick")
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats   1.0.0      ✓ stringr   1.5.1
## ✓ ggplot2   3.5.2      ✓ tibble    3.2.1
## ✓ lubridate 1.9.4      ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(ggplot2)
library(grid)
library(dplyr)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor
```

```
library(colorspace)
library(magick)
```

```
## Linking to ImageMagick 6.9.12.98
## Enabled features: cairo, freetype, fftw, ghostscript, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fontconfig, x11
```

0. Load image & dimensions

```
# Load original image
img <- image_read("images/original.png")
info <- image_info(img)
img_width  <- info$width
img_height <- info$height

# Print original image
gg1 <- ggplot() +
  annotation_custom(
    rasterGrob(img, width=unit(1,"npc"), height=unit(1,"npc")),
    xmin=0, xmax=img_width, ymin=0, ymax=img_height
  ) +
  coord_fixed(
    ratio = 1,
    xlim  = c(0, img_width),
    ylim  = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  theme(
    panel.background = element_rect(fill = "white", color = NA),
    plot.background  = element_rect(fill = "white", color = NA)
  )
print(gg1)
```



1. Mountain (sky) polygon

```

# Define the Mountain curves
line <- data.frame(x = c(1158,1225), y = c(163,167))

# Use Bezier function for better smooth
bezier <- function(p0, p1, p2, n = 100) {
  t <- seq(0, 1, length.out = n)
  data.frame(
    x = (1 - t)^2 * p0[1] + 2*(1 - t)*t * p1[1] + t^2 * p2[1],
    y = (1 - t)^2 * p0[2] + 2*(1 - t)*t * p1[2] + t^2 * p2[2]
  )
}

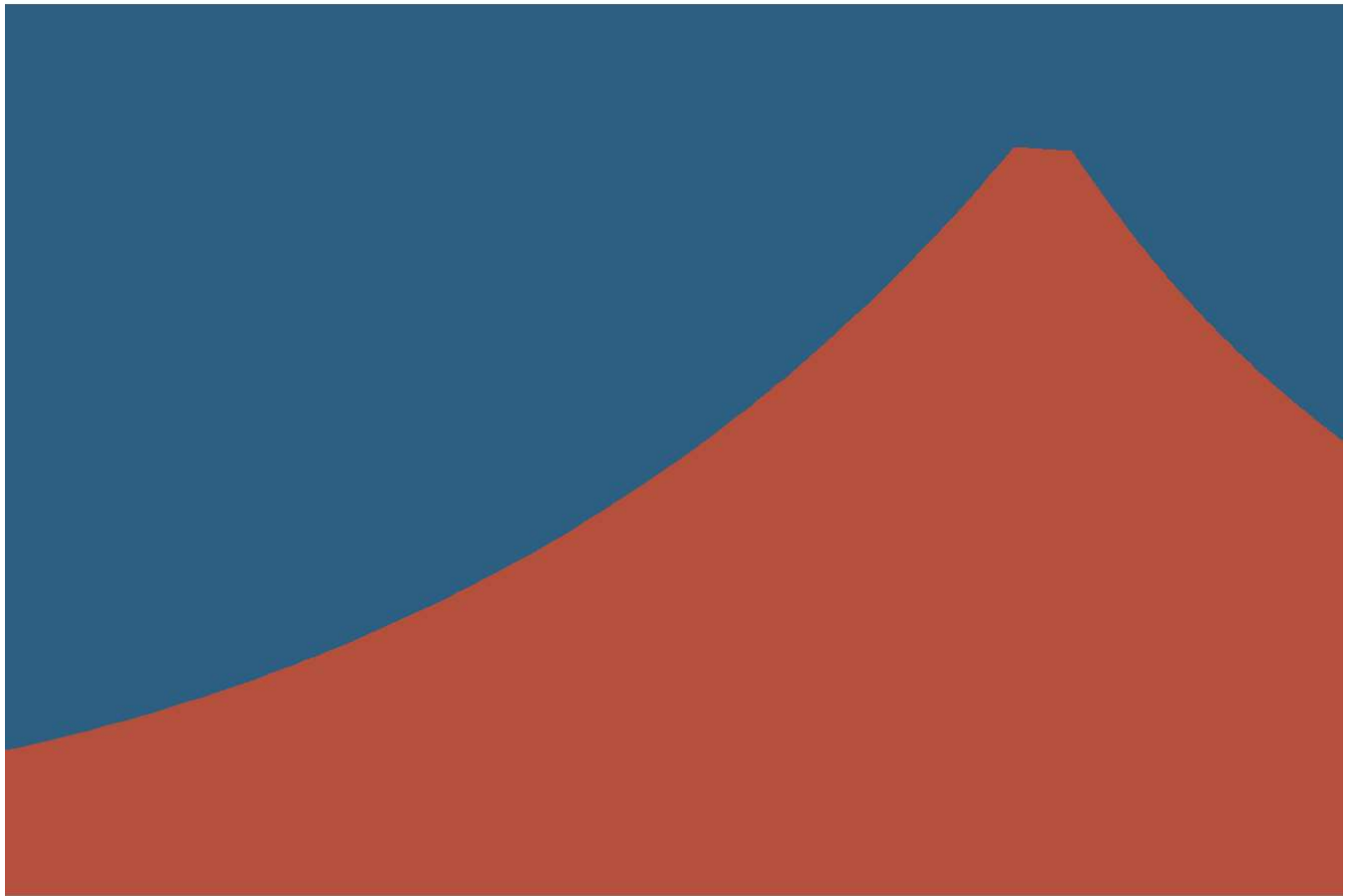
curve1 <- bezier(c(0, 856), c(715, 700), c(1158, 163), n = 100)
curve2 <- bezier(c(1225,167), c(1360,370), c(1536,500), n = 100)
mount_outline <- bind_rows(curve1, line, curve2)
mount_poly <- bind_rows(
  mount_outline,
  data.frame(x = c(max(mount_outline$x), min(mount_outline$x)), y = c(0, 0))
)

# The rest of the area will be the sky
sky_layer <- geom_polygon(
  data = mount_poly,
  aes(x = x, y = img_height - y),
  fill = "#316283",
  color = NA
)

gg2 <- ggplot() +
  #the layers in order
  sky_layer +

  coord_fixed(
    ratio = 1,
    xlim = c(0, img_width),
    ylim = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  # Simple color for mountain for now
  theme(
    panel.background = element_rect(fill = "#b5533c", color = NA),
    plot.background = element_rect(fill = "white", color = NA)
  )
#print(gg1)
print(gg2)

```



2. Forest polygon

```
# Define the Forest Area

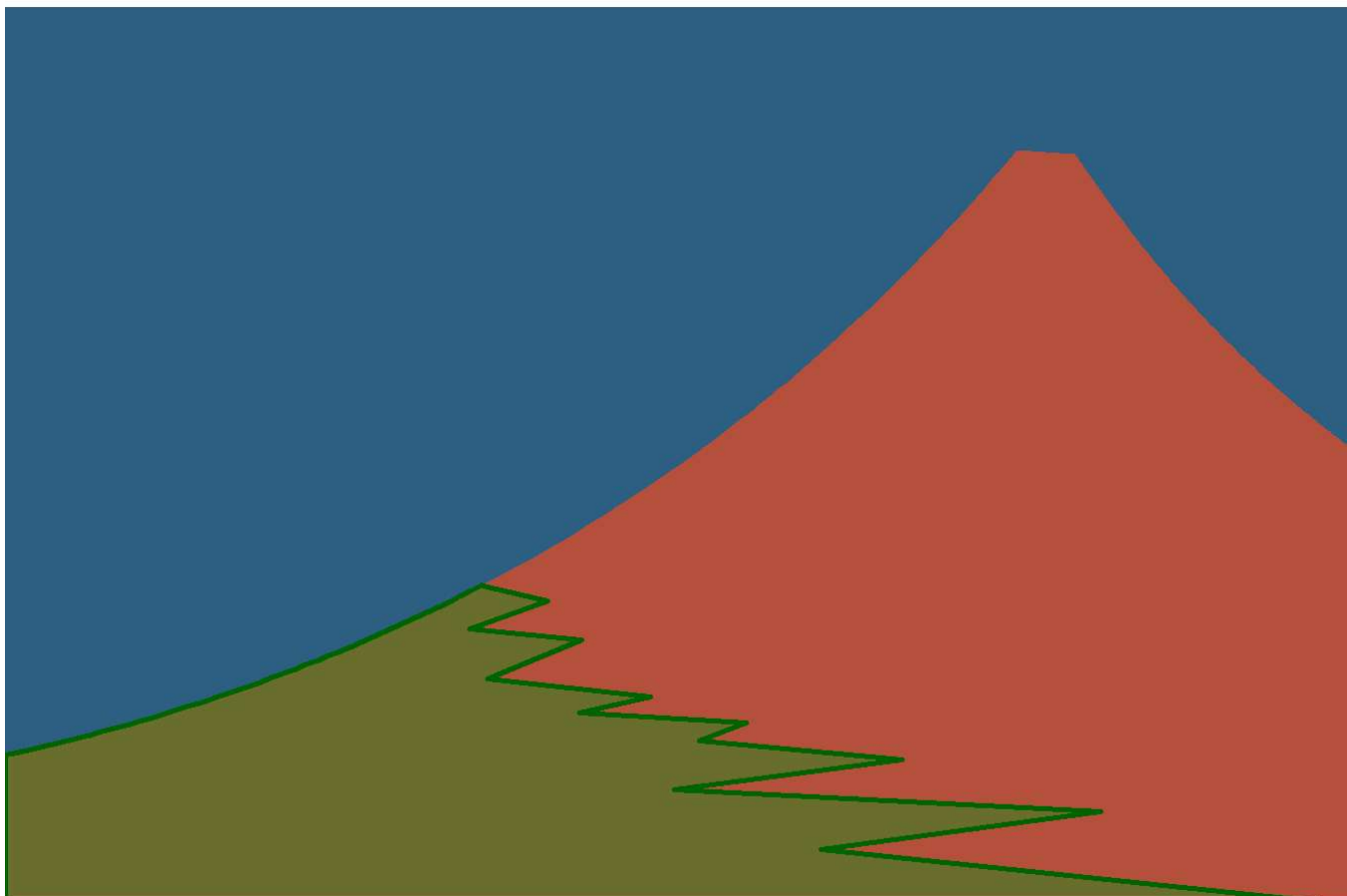
forest_pts <- data.frame(
  x = c(552,621,532,660,552,739,658,849,795,1027,766,1254,934,1500),
  y = c(663,679,711,724,769,789,808,819,840,861,895,921,964,1024)
)
dists      <- sqrt((curve1$x - forest_pts$x[1])^2 + (curve1$y - forest_pts$y[1])^2)
idx_min    <- which.min(dists)
mount_seg  <- curve1[1:idx_min, ]
edge       <- data.frame(x = c(1449, 0, 0), y = c(1024, 1024, 856))
forest_poly <- rbind(mount_seg, forest_pts, edge)

forest_layer <- geom_polygon(
  data = forest_poly,
  aes(x = x, y = img_height - y),
  fill = "forestgreen",
  alpha = 0.5,
  color = "darkgreen",
  size = 1
)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
gg2 <- ggplot() +
  #the layers in order
  sky_layer +
  forest_layer+

  coord_fixed(
    ratio = 1,
    xlim = c(0, img_width),
    ylim = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  theme(
    panel.background = element_rect(fill = "#b5533c", color = NA),
    plot.background = element_rect(fill = "white", color = NA)
  )
#print(gg1)
print(gg2)
```



3. Clouds as points (from cloud.csv)

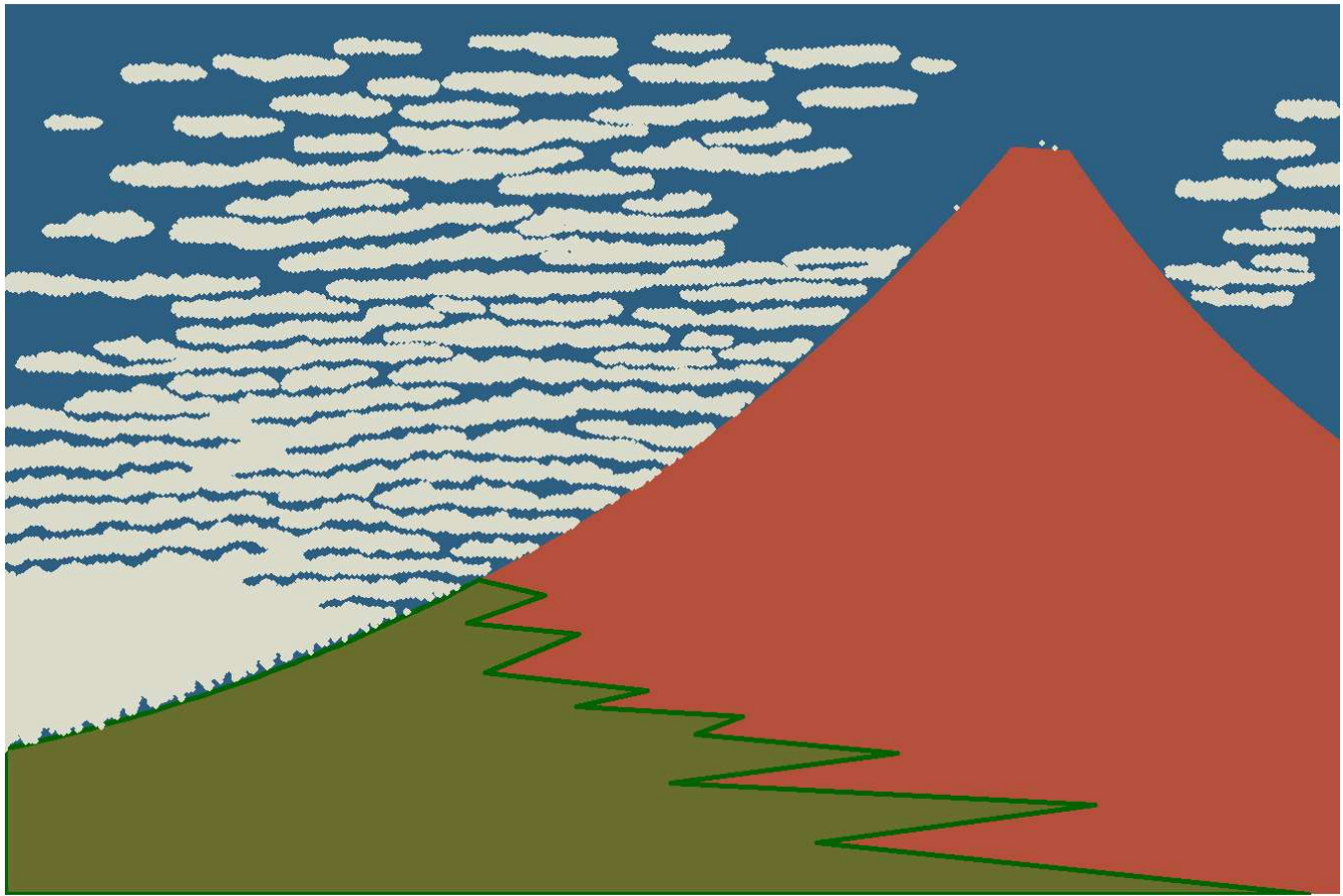
```
# The coordinate data of clouds are extracted using extract_cloud_data.py

cloud_df <- read.csv("data/cloud.csv")
cloud_layer <- geom_point(
  data = cloud_df,
  aes(x = x, y = y),
  shape = 21, fill = "#e0dccc", color = "#e0dccc", size = 0.66,
  inherit.aes = FALSE
)

gg2 <- ggplot() +
  #the layers in order
  sky_layer +
  forest_layer+
  cloud_layer +

  coord_fixed(
    ratio = 1,
    xlim = c(0, img_width),
    ylim = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  theme(
    panel.background = element_rect(fill = "#b5533c", color = NA),
    plot.background = element_rect(fill = "white", color = NA)
  )

#print(gg1)
print(gg2)
```

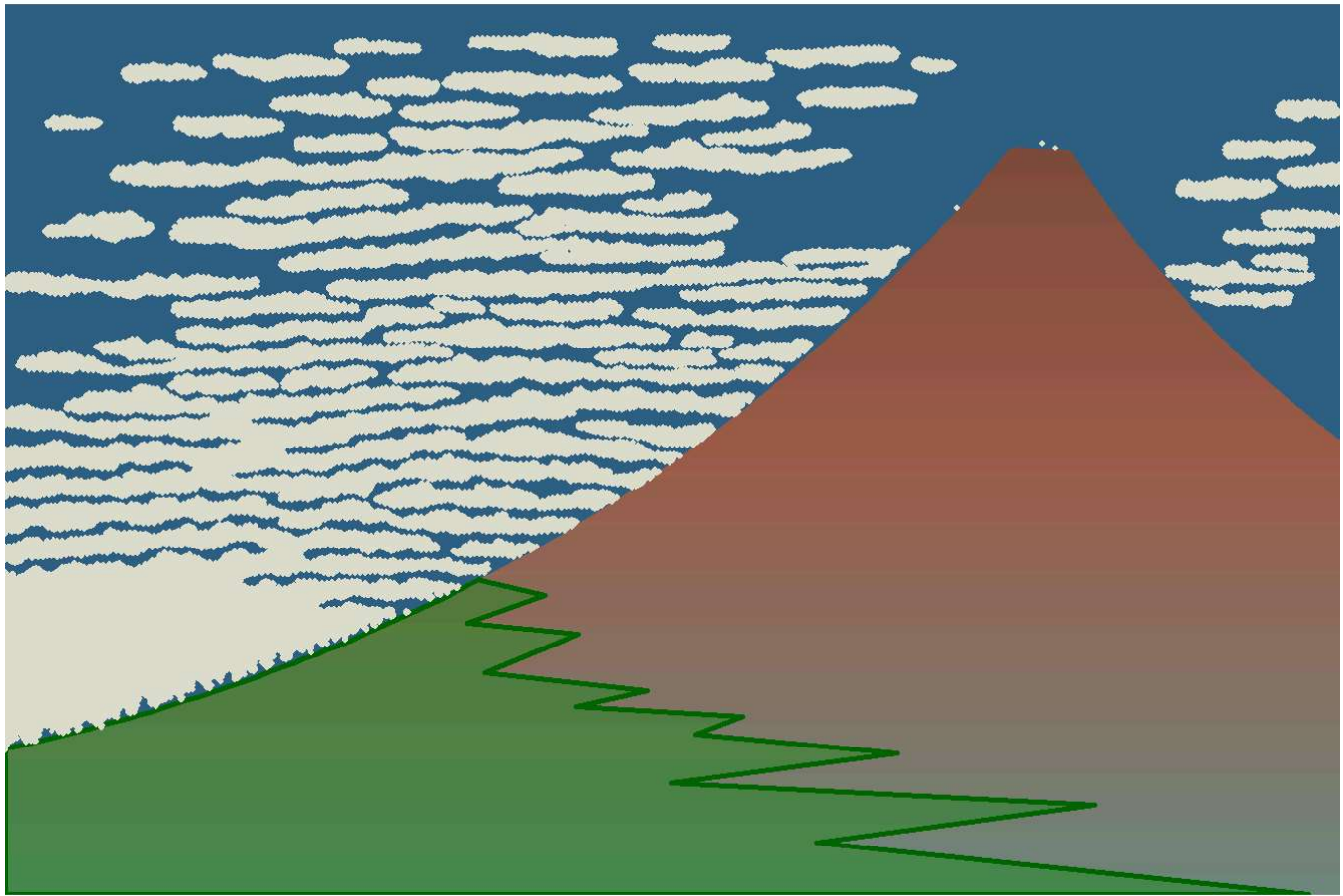



```
# 4. Gradient raster for the mountain
lower_color  <- "#70897f" # foot
middle_color <- "#9e5d4a" # mid
highest_color <- "#714739" # peak
grad_colors  <- colorRampPalette(
  c(highest_color, middle_color, lower_color)
)(img_height)
grad_raster <- as.raster(matrix(grad_colors, ncol = 1, byrow = TRUE))

gg2 <- ggplot() +
  annotation_raster(
    grad_raster,
    xmin = 0, xmax = img_width,
    ymin = 0, ymax = img_height
  ) +

  #the layers in order
  sky_layer +
  forest_layer+
  cloud_layer +

  coord_fixed(
    ratio = 1,
    xlim = c(0, img_width),
    ylim = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  theme(
    panel.background = element_rect(fill = "#b5533c", color = NA),
    plot.background  = element_rect(fill = "white",   color = NA)
  )
#print(gg1)
print(gg2)
```



5. Trees from CSV

```

# The coordinate data of trees are extracted using extract_tree_data.py

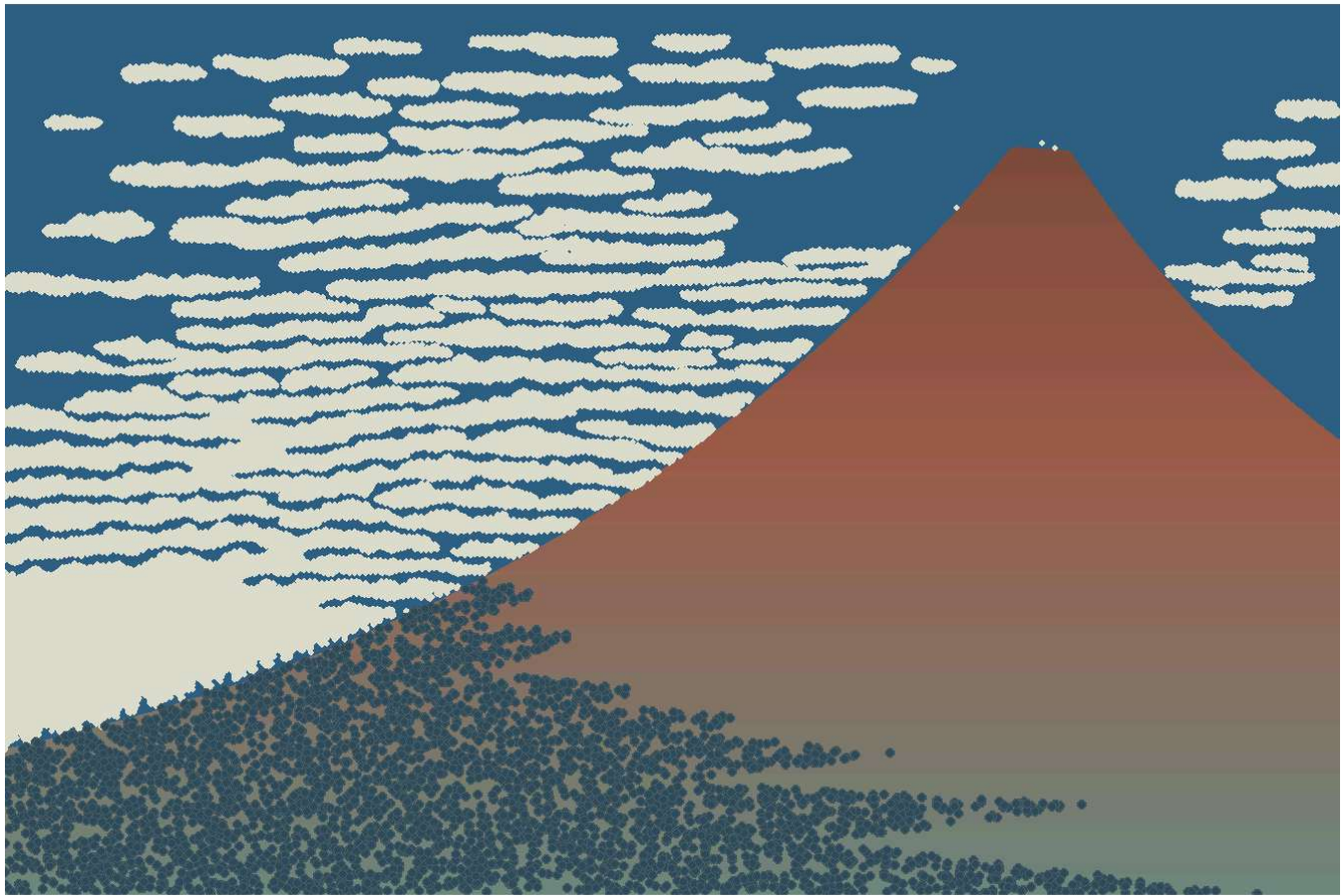
trees_df <- read.csv("data/trees.csv")
tree_layer <- geom_point(
  data = trees_df,
  aes(x = x, y = y),
  shape = 21, fill = "#304b5e", color = "#4d6064", size = 1.5,
  inherit.aes = FALSE
)

gg2 <- ggplot() +
  annotation_raster(
    grad_raster,
    xmin = 0, xmax = img_width,
    ymin = 0, ymax = img_height
  ) +

  #the layers in order, the forest layer is no longer needed
  sky_layer +
  #forest_layer+
  cloud_layer +
  tree_layer +

  coord_fixed(
    ratio = 1,
    xlim = c(0, img_width),
    ylim = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  theme(
    panel.background = element_rect(fill = "#b5533c", color = NA),
    plot.background = element_rect(fill = "white", color = NA)
  )
#print(gg1)
print(gg2)

```



6. Snow from CSV

```
# The coordinate data of snow are extracted using extract_snow_data.py
```

```
snow_df <- read.csv("data/snow.csv")
snow_layer <- geom_point(
  data = snow_df,
  aes(x = x, y = y),
  shape = 21, fill = "#ebe3d4", color = "#ebe3d4", size = 1.25,
  inherit.aes = FALSE
)

gg2 <- ggplot() +
  annotation_raster(
    grad_raster,
    xmin = 0, xmax = img_width,
    ymin = 0, ymax = img_height
  ) +

  #the layers in order
  sky_layer +
  #forest_layer+
  cloud_layer +
  tree_layer +
  snow_layer +

  coord_fixed(
    ratio = 1,
    xlim = c(0, img_width),
    ylim = c(0, img_height),
    expand= FALSE
  ) +
  theme_void() +
  theme(
    panel.background = element_rect(fill = "#b5533c", color = NA),
    plot.background = element_rect(fill = "white", color = NA)
  )
#print(gg1)
print(gg2)
```

