# zone_data

Haseeb Raza

2024-09-18

## Importing Data from Excel

```r
library(readxl)
zone_main_data <-  read_excel("zone wise data 10-12-2023.xlsx",
    sheet = "zone wise general sheet", range = "B1:FM50")
View(zone_main_data)
```

zone_main_data <- read_excel(…): Reads data from an Excel file and stores it in the dataframe zone_main_data.

"zone wise data 10-12-2023.xlsx": Specifies the Excel file's name. sheet = "zone wise general sheet": Indicates that the data is being read from a sheet named "zone wise general sheet." range = "B1:FM50": Specifies the range of cells to read, which includes columns B to FM and rows 1 to 50. View(zone_main_data): Opens a viewer window to visually inspect the contents of the dataframe zone_main_data in RStudio.

```r
library(readxl)
zone_raw_data <-  read_excel("zone wise data 10-12-2023.xlsx",
    sheet = "zone wise general sheet", range = "A5:FI54")
View(zone_raw_data)
```

zone_raw_data <- read_excel(…): Reads data from an Excel file and stores it in the dataframe zone_raw_data.

"zone wise data 10-12-2023.xlsx": Specifies the name of the Excel file to be read. sheet = "zone wise general sheet": Indicates the sheet "zone wise general sheet" from which the data is being read. range = "A5:FI54": Specifies the range of cells (from column A, row 5 to column FI, row 54) to be imported from the sheet. View(zone_raw_data): Opens a viewer window in RStudio to inspect the contents of zone_raw_data visually.

```r
library(readxl)
zone_data <- read_excel("zone wise data 10-12-2023.xlsx",
    sheet = "zone wise general sheet", range = "B61:AB110",
    col_types = c("text", "numeric", "numeric",
        "numeric", "numeric", "numeric",
        "numeric", "numeric", "numeric",
        "numeric", "numeric", "numeric",
        "numeric", "text", "numeric", "numeric",
        "numeric", "numeric", "numeric",
        "numeric", "numeric", "numeric",
        "numeric", "numeric", "numeric",
        "numeric", "numeric"))
View(zone_data)
```

library(readxl): Loads the readxl package to enable reading Excel files.

zone_data <- read_excel(…): Reads data from an Excel file and stores it in the dataframe zone_data.

"zone wise data 10-12-2023.xlsx": Specifies the Excel file name. sheet = "zone wise general sheet": Indicates the sheet from which the data is being read. range = "B61:AB110": Specifies the cell range to read (from column B, row 61 to column AB, row 110). col_types = c(…): Specifies the data types for each column in the selected range. "text" indicates a column with character data. "numeric" indicates columns with numerical data (like integers or decimals). The col_types argument provides a detailed format for each column in the specified range, ensuring that the data is read correctly.

View(zone_data): Opens the dataframe in a viewer window in RStudio for visual inspection.

```
library(readxl)
zone_info_data <- read_excel("zone wise data 10-12-2023.xlsx",
    sheet = "zone wise general sheet", range = "B116:EJ165")
View(zone_info_data)
```

library(readxl): Loads the readxl package, which allows you to read data from Excel files.

zone_info_data <- read_excel(…): Reads the data from an Excel file and stores it in the zone_info_data dataframe.

"zone wise data 10-12-2023.xlsx": Specifies the name of the Excel file to read (located in the working directory). sheet = "zone wise general sheet": Selects the specific sheet in the Excel file from which the data should be read. range = "B116:EJ165": Specifies the range of cells (from column B, row 116 to column EJ, row 165) to be read from the sheet. View(zone_info_data): Opens a spreadsheet-like view of the loaded dataframe zone_info_data, allowing you to visually inspect the data in the RStudio viewer.

# Data Wrangling and Plotting using tidyverse(), ggplot2() & Allied Packages

```
library(tidyverse)
# summary(zone_data)
# glimpse(zone_data)
```

summary(zone_data) gives statistical summaries for numeric columns and count summaries for text columns. glimpse(zone_data) provides a concise, structured view of the data's columns and types.

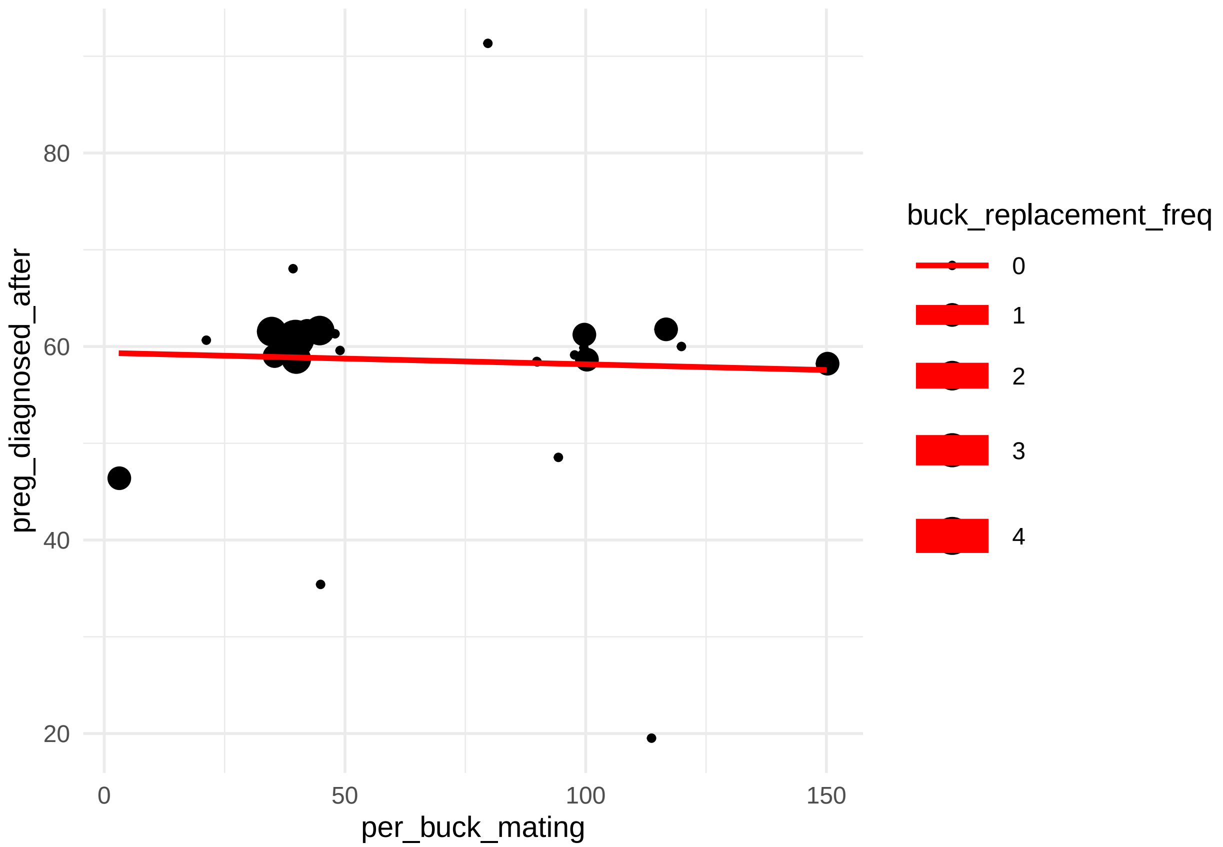
```
library(tidyverse)
zone_data <- zone_data %>%  rename("farm_location" = "2. Farm Location:",
                      "zones" = "Zones",
                      "numb_ani_herd" = "11. No. of animals in herd:",
                      "male_animals_numb" = "14. Number of Male Animals?",
                      "female_animals_numb" = "15. Number of Female Animals?",
                      "young_stock" = "16. Number of young stock?" ,
                      "male_young_stock" = "17. Male young stock less than six month?",
                      "female_young_stock" = "18.Female young stock less than six month?",
                      "breeding_stock" = "19. Breeding stock?",
                      "female_breeding_stock" = "20.No of Female breeding stock?",
                      "male_breeding_stock" = "21. No of male breeding stock?" ,
                      "buck_replacement_freq" = "31. Buck replacement frequency within the season?",
                      "bucks_used_in_season" = "32.How many bucks are used for breeding in a given seas
                      "month_gestation" = "43. If yes, then at which month of gestation?",
                      "ages_at_first_service" = "44. Breeding Pattern: Age at 1st Service?",
                      "ave_milk_prod" = "55. Average Milk Production",
                      "per_buck_mating" = "58. Per Buck Number of Mating in given Season:",
                      "preg_diagnosed_after" = "60. After how many days of mating pregnancy diagnosis pe
                      "bred_duration" = "69. How long after parturition  female is  bred?",
                      "buck_remaind_sep" = "76. How many days breeding buck remained separated  from fe
                      "kids_mortality_summer" = "83. What was the mortality rate of kids in summer seas
                      "kids_mortality_winter" = "84. What was the mortality rate of Kids in winter seas
                     "ages_at_castration" = "91. If yes, then at what age castration is performed?",
                     "weaning_age" = "96. At what age Weaning done?",
                    "ages_at_fodder" = "97. At what age fodder is offered?",
                    "buck_replaced_after" = "29. After how many seasons buck is replaced with new one?"
                    "ave_milk_prod_new" = "87. Average milk production?")
```

The rename function in dplyr is used to rename columns in a data frame.

```
library(tidyverse)
 zone_data <- zone_data %>%
  filter(farm_location %in% c("Faislabad","BWP","Rawalpindi","Hafizabad",  "Vehari" ) ) %>%
  mutate(farm_location = recode( farm_location, "Faislabad" = "Faisalabad",
                                "BWP" = "Bahawalpur"))
```

Initially, it filters the dataset to retain only the rows where the farm_location column matches one of five specified values: “Faislabad”, “BWP”, “Rawalpindi”, “Hafizabad”, and “Vehari”. Subsequently, it uses the mutate and recode functions to correct and standardize the values in the farm_location column, changing “Faislabad” to “Faisalabad” and “BWP” to “Bahawalpur”. This process ensures that the data is both filtered and cleaned, making it ready for further analysis.

```
library(tidyverse)
library(ggplot2)
ggplot(data = zone_data, aes(x=per_buck_mating, y=preg_diagnosed_after, size = buck_replacement_freq, pe
```



It creates a scatter plot where per_buck_mating is plotted on the x-axis and preg_diagnosed_after on the y-axis, with point sizes representing buck_replacement_freq. The plot is enhanced with jitter to reduce overlap and includes a linear trend line, colored red, without a confidence interval, all styled using the minimal theme.

## transformation to long formate

```r
library(tidyverse)
zone_data <- zone_data %>%
  rename("male"= "male_animals_numb", "female" = "female_animals_numb") %>%
  pivot_longer(cols = c(male, female ), names_to = "gender", values_to = "number_of_animals" )
```

It begins by renaming the columns male_animals_numb to male and female_animals_numb to female. Following this, it reshapes the data from a wide format to a long format using the pivot_longer function, consolidating the male and female columns into two new columns: gender, which indicates the animal gender, and number_of_animals, which provides the corresponding counts. This transformation makes the dataset more suitable for subsequent analysis and visualization.

```r
library(tidyverse)
zone_data <- zone_data %>%
  rename("total_number" = "numb_ani_herd",
         "total_young_stock" = "young_stock",
         "male" = "male_young_stock",
         "female" = "female_young_stock",
         ) %>%
  pivot_longer(cols = c(male, female), names_to = "stock_gender", values_to = "number_of_stock")
```

it renames several columns: numb_ani_herd to total_number, young_stock to total_young_stock, male_young_stock to male, and female_young_stock to female. Then, it employs the pivot_longer function to convert the dataset from wide to long format, merging the male and female columns into two new columns: stock_gender, indicating the gender of the young stock, and number_of_stock, showing the corresponding counts. This transformation facilitates easier analysis and visualization of the data.

```
library(tidyverse)
 zone_data <- zone_data %>%
  rename("total_breeding_stock" = "breeding_stock",
         "female" = "female_breeding_stock",
         "male" = "male_breeding_stock",
         ) %>%
  pivot_longer(cols = c(male, female), names_to = "breeder_gender", values_to = "number_of_breeding_sto
```

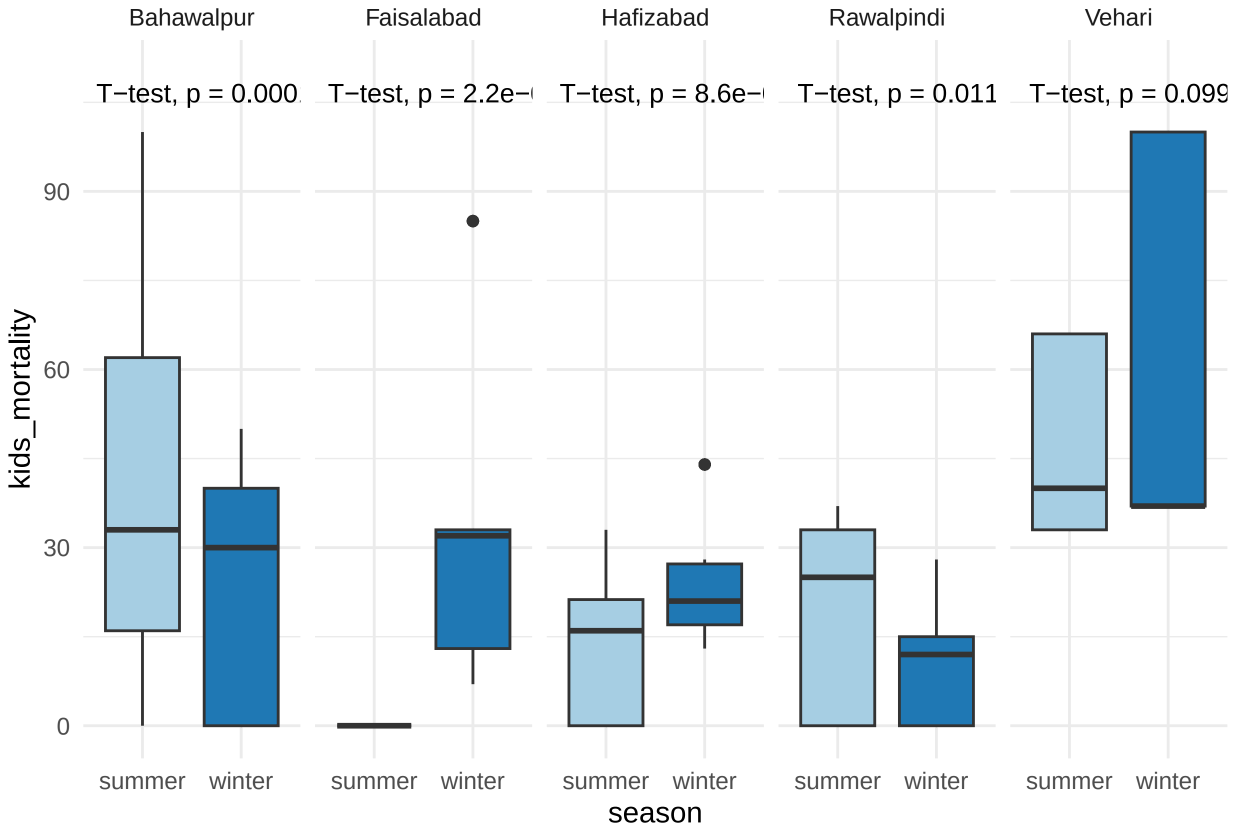
First, it renames several columns, changing "breeding_stock" to "total_breeding_stock", "female_breeding_stock" to "female", and "male_breeding_stock" to "male". Then, it reshapes the data from a wide format to a long format using pivot_longer(). Specifically, the columns "male" and "female" are consolidated into a single column named "breeder_gender", with the values placed into a new column called "number_of_breeding_stock". This transformation makes the data more suitable for analysis by categorizing the breeding stock by gender.

```
library(tidyverse)
 zone_data <- zone_data %>% rename("summer" = "kids_mortality_summer",
                     "winter" = "kids_mortality_winter") %>%
  pivot_longer(cols = c(summer, winter), names_to = "season",
               values_to = "kids_mortality")
```

it takes the columns "summer" and "winter", which contain mortality data for kids (young animals), and combines them into a single column named "season". The corresponding values from these columns are placed into a new column called "kids_mortality". This transformation allows for easier analysis of the mortality data by season, with both summer and winter data represented under a unified structure.

## Visualizing Kids Mortality in Summer & Winter

```
library(tidyverse)
library(ggplot2)
library(ggpubr)
y_max <- max(zone_data$kids_mortality)
zone_data %>%
  group_by(farm_location) %>%
  ggplot(aes(x=season , y = kids_mortality, fill = season)) + geom_boxplot() +  facet_wrap(~farm_locati
  coord_cartesian(ylim = c(0, 110)) + scale_fill_brewer(palette = "Paired") +
  theme(legend.position = "none") +
  geom_smooth(se=FALSE , method = "lm")
```



```
# + geom_violin(position = "dodge", scale = "area")


# geom_jitter(position = position_jitter(width = 0.1, height = 0), size = 1, alpha = 0.6)
```

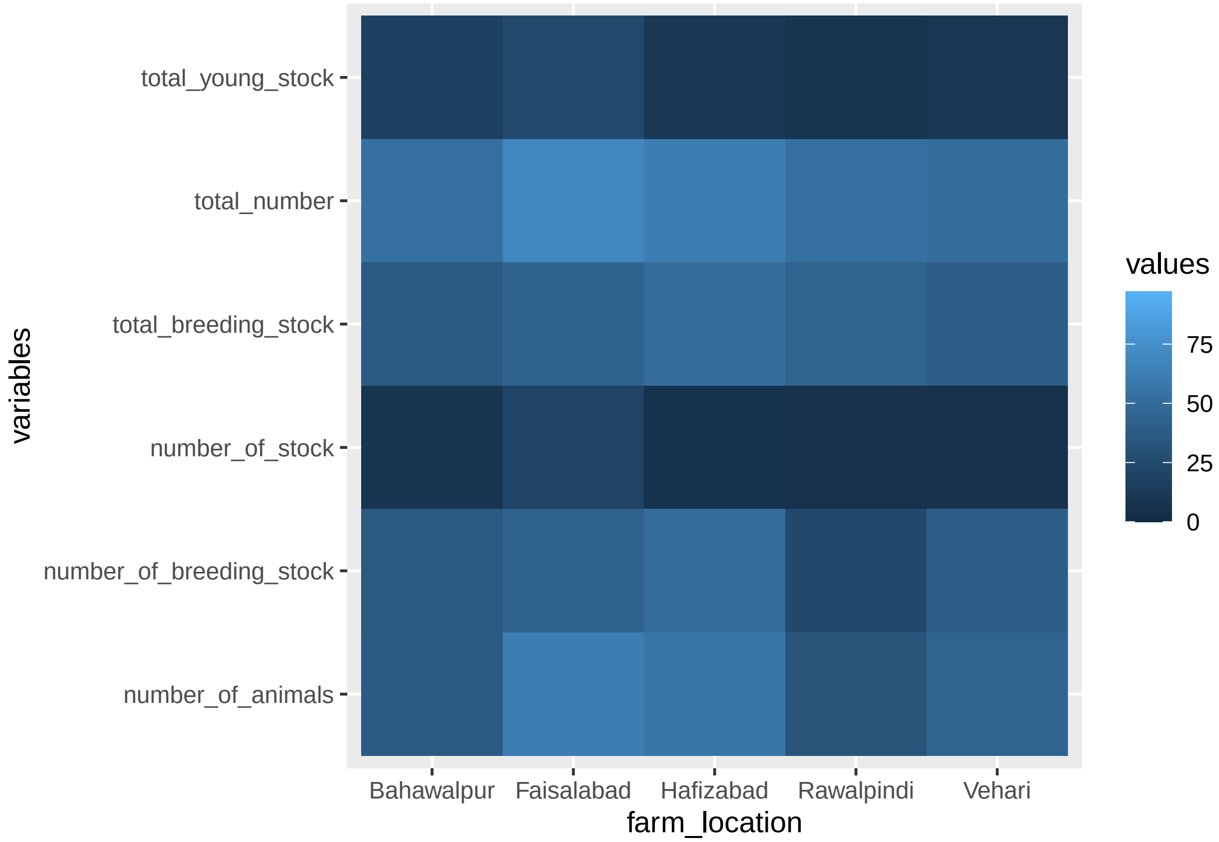
After grouping the data by farm_location, it plots the kids_mortality values on the y-axis and the season on the x-axis, filling the boxplots by season. The facet_wrap() function organizes the plots into a single row for each farm location. A t-test comparison (stat_compare_means) is applied between seasons, with the results displayed above the boxplots. The plot limits are set using coord_cartesian() to cap the y-axis at 110, while the color palette is adjusted with scale_fill_brewer(). A linear trend line is also added using geom_smooth(), without showing the confidence interval, and a minimalist theme is applied with the legend removed.

### Preparing Data for a Heatmap using ggplot2() Package

```
library(tidyverse)
# library(ggplot2)
# library(reshape)


heatmap_1 <- zone_data %>% select(., farm_location, total_number, total_young_stock, total_breeding_sto
             number_of_animals, number_of_stock,
                       number_of_breeding_stock)
```

```r
library(tidyverse)
library(ggplot2)
 heatmap_1 %>%
  pivot_longer(cols = -farm_location, names_to = "variables" , values_to = "values") %>%
  ggplot(aes(farm_location, variables, fill = values)) + geom_tile() + scale_color_gradient(low = "yell
```



The first code chunk creates a subset of the zone_data dataset called heatmap_1, which selects specific columns including farm_location, total_number, total_young_stock, total_breeding_stock, number_of_animals, number_of_stock, and number_of_breeding_stock. The second chunk reshapes this subset into a long format using pivot_longer(), where all columns except farm_location are consolidated under a new column named variables, with their corresponding values stored in a new values column. This reshaped data is then used to create a heatmap with ggplot2, where the x-axis represents farm_location, the y-axis represents variables, and the fill color intensity, ranging from yellow to red, reflects the magnitude of the values using geom_tile().

## Data Part 2

```r
library(readxl)
zone_info_data <- read_excel("zone wise data 10-12-2023.xlsx",
    sheet = "zone wise general sheet", range = "B116:EJ165")
View(zone_info_data)
```

```
 # colnames(zone_info_data)
```

# Data Transformation & Analysis

```
library(tidyverse)
zone_info_data %>%
  summarise(names = colnames(select(.,  starts_with("Vq"))))
```

```
## # A tibble: 69 x 1
##    names
##    <chr>
##  1 Vq2
##  2 Vq4
##  3 Vq6
##  4 Vq7
##  5 Vq8
##  6 Vq9
##  7 Vq10
##  8 Vq12
##  9 Vq13
## 10 Vq22
## # i 59 more rows
```

```
# %>%
 # print(n=69)
```

```
library(tidyverse)
zone_info_data %>%
  select(., - starts_with("Vq"))
```

```
## # A tibble: 49 x 70
##    ‘2. Farm Location:‘ Zones ‘4.Education of farmer?‘ 5.For how many years you~1
##    <chr>               <dbl> <chr>                                         <dbl>
##  1 Faislabad               8 Illiterate                                       40
##  2 Faislabad               8 Illiterate                                       30
##  3 Faislabad               8 Matric                                            7
##  4 Faislabad               8 Primary                                          20
##  5 Faislabad               8 Illiterate                                       50
##  6 BWP                     5 Illiterate                                       40
##  7 BWP                     5 Graduation                                       45
##  8 BWP                     5 Illiterate                                       40
##  9 BWP                     5 Primary                                          15
## 10 BWP                     5 Primary                                          12
## # i 39 more rows
## # i abbreviated name: 1: ‘5.For how many years you are in this business?‘
## # i 66 more variables: ‘6.Which preferable breed of  goat do you have?‘ <chr>,
## #   ‘7. Record Keeping?‘ <chr>, ‘8. Record related to‘ <chr>,
## #   ‘9.Do you have separate housing for does/bucks/kids?‘ <chr>,
## #   ‘10. If not separate housing what is the reason?‘ <chr>,
## #   ‘12. Production System?‘ <chr>, ‘13. Management system?‘ <chr>, ...
```

```
# updating zone_info_data

zone_info_data <- zone_info_data %>%
  select(., - starts_with("Vq"))
```

The first code chunk filters the zone_info_data dataset by selecting all columns except those that start with the prefix “Vq”, effectively excluding them from the result. The second code chunk performs the same operation but additionally updates the original zone_info_data dataset by reassigning it to the filtered version, thereby permanently removing columns that start with “Vq”.

```
zone_info_data <- zone_info_data %>%
  rename( "farm_location" = "2. Farm Location:",
          "zones" = "Zones",
          "edu_farmer" = "4.Education of farmer?",
          "years_in_business" = "5.For how many years you are in this business?",
          "pref_breed" = "6.Which preferable breed of  goat do you have?")
```

```
zone_info_data <- zone_info_data  %>% filter(farm_location %in% c("Faislabad","BWP","Rawalpindi","Hafiza
  mutate(farm_location = recode( farm_location, "Faislabad" = "Faisalabad",
                                "BWP" = "Bahawalpur"))
```

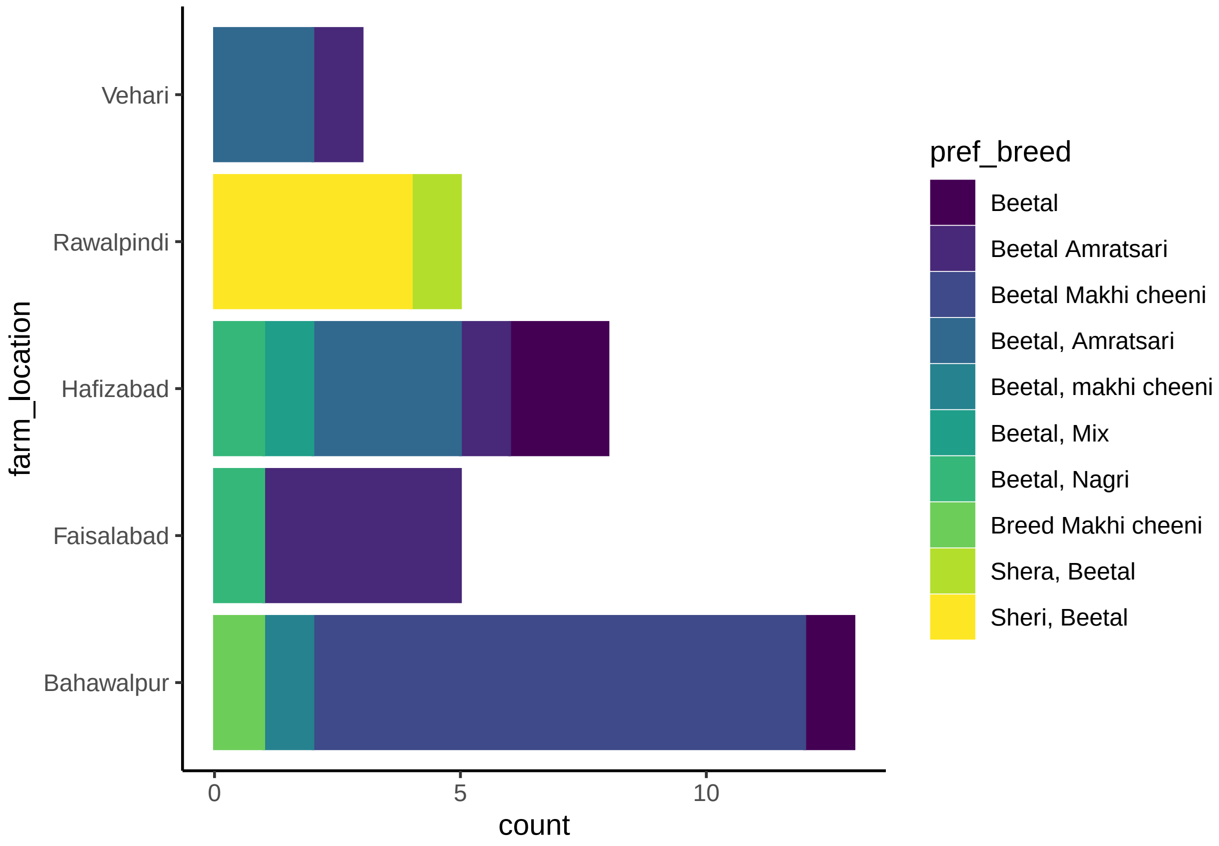
```
zone_info_data %>%
  mutate( edu_farmer = recode( edu_farmer ,"illeterate" = "Illiterate")) %>%
  select(., c(farm_location : pref_breed))
```

```
## # A tibble: 34 x 5
##    farm_location zones edu_farmer years_in_business pref_breed
##    <chr>         <dbl> <chr>                  <dbl> <chr>
##  1 Faisalabad        8 Illiterate                40 Beetal Amratsari
##  2 Faisalabad        8 Illiterate                30 Beetal, Nagri
##  3 Faisalabad        8 Matric                     7 Beetal Amratsari
##  4 Faisalabad        8 Primary                   20 Beetal Amratsari
##  5 Faisalabad        8 Illiterate                50 Beetal Amratsari
##  6 Bahawalpur        5 Illiterate                40 Beetal, makhi cheeni
##  7 Bahawalpur        5 Graduation                45 Beetal
##  8 Bahawalpur        5 Illiterate                40 Beetal Makhi cheeni
##  9 Bahawalpur        5 Primary                   15 Beetal Makhi cheeni
## 10 Bahawalpur        5 Primary                   12 Beetal Makhi cheeni
## # i 24 more rows
```

```
library(tidyverse)
library(ggplot2)
library(esquisse)


library(ggplot2)

ggplot(zone_info_data) +
 aes(x = farm_location, fill = pref_breed, colour = pref_breed) +
 geom_bar() +
 scale_fill_viridis_d(option = "viridis", direction = 1) +
 scale_color_viridis_d(option = "viridis",
 direction = 1) +
 coord_flip() +
 theme_classic()
```



```
library(tidyverse)
library(ggplot2)

zone_info_data <- zone_info_data %>%
  rename("sepr_housing" = "9.Do you have separate housing for does/bucks/kids?",
         "reason" = "10. If not separate housing what is the reason?",
         "housing_winter" = "22. Housing in winter?",
         "housing_summer" = "23. Housing in summer?" )
```

```
library(esquisse)
zone_info_data %>%
  select(.,farm_location, sepr_housing, reason, housing_winter, housing_summer )
```

```
## # A tibble: 34 x 5
##    farm_location sepr_housing reason            housing_winter    housing_summer
##    <chr>         <chr>        <chr>             <chr>             <chr>
##  1 Faisalabad    No           Nil               Make shift Arran~ Make shift Ar~
##  2 Faisalabad    No           Nil               Make shift Arran~ Make shift Ar~
##  3 Faisalabad    Yes          Nil               Make shift Arran~ Make shift Ar~
##  4 Faisalabad    Yes          For kids and male Make shift Arran~ Make shift Ar~
##  5 Faisalabad    Yes          For kids and male Proper Housing    Proper Housing
##  6 Bahawalpur    No           Nil               Make shift Arran~ Make shift Ar~
##  7 Bahawalpur    Yes          Nil               Make shift Arran~ Make shift Ar~
##  8 Bahawalpur    Yes          Male              Make shift Arran~ Make shift Ar~
##  9 Bahawalpur    No           Nil               Make shift Arran~ Make shift Ar~
## 10 Bahawalpur    No           Nil               Make shift Arran~ Make shift Ar~
## # i 24 more rows
```