

ASSIGNMENT – 4

Name: M Gyanada Chowdary

Reg.no: 21BCE7727

Slot: L39+L40

1. Create a Scatterplot Matrix in R.

GYANADA

21BCE7727

L39+L40

LAB 4

1. Create a ScatterPlot Matrix in R.

```
In [67]: library(ggplot2)
library(GGally)
```

```
In [68]: data("iris")
df <- data.frame(iris)
```

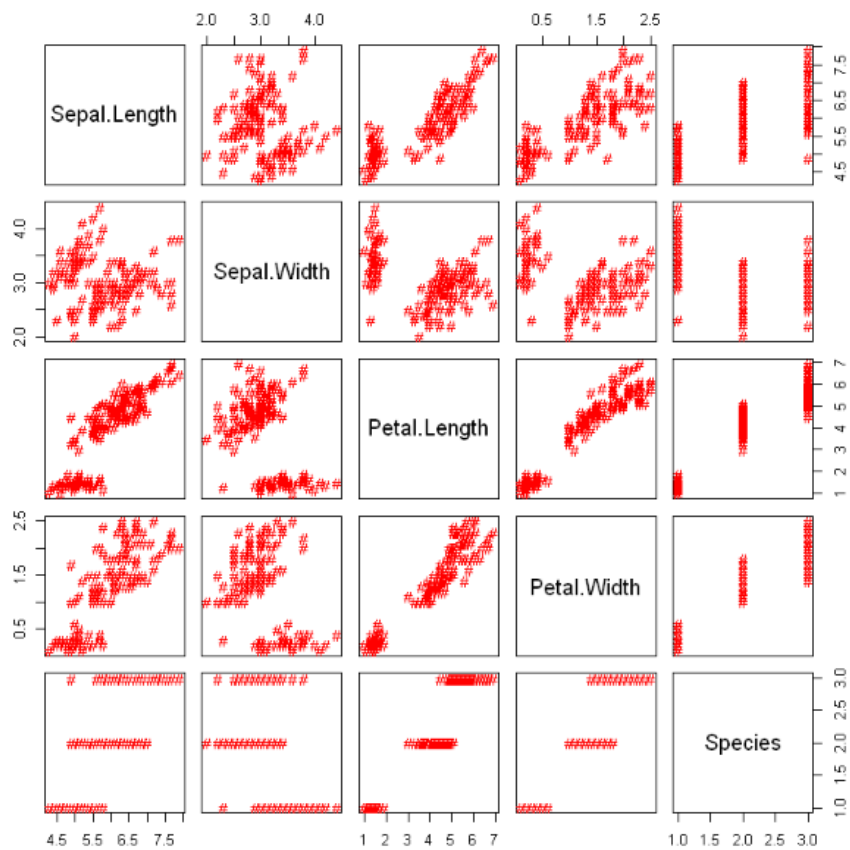
```
In [69]: head(iris)
```

A data.frame: 6 × 5

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

```
In [70]: plot(df, pch=35, cex=1, col='red')
```

0.0 0.5 1.0 1.5 2.0



2. Draw a graph matrix for Image Segmentation multivariate dataset (available in UCI repository) ¶

• Use customized functions for the panels to Scatter plot with correlations in the upper triangle,

smoothing lines in the lower triangle, and histograms on the diagonal.

```
In [71]: library(datasets)
library(GGally)

In [72]: file <- "C:/Users/gyanada/Downloads/image+segmentation/segmentation.data"

In [74]: segmentation <- read.csv(file, header = FALSE)

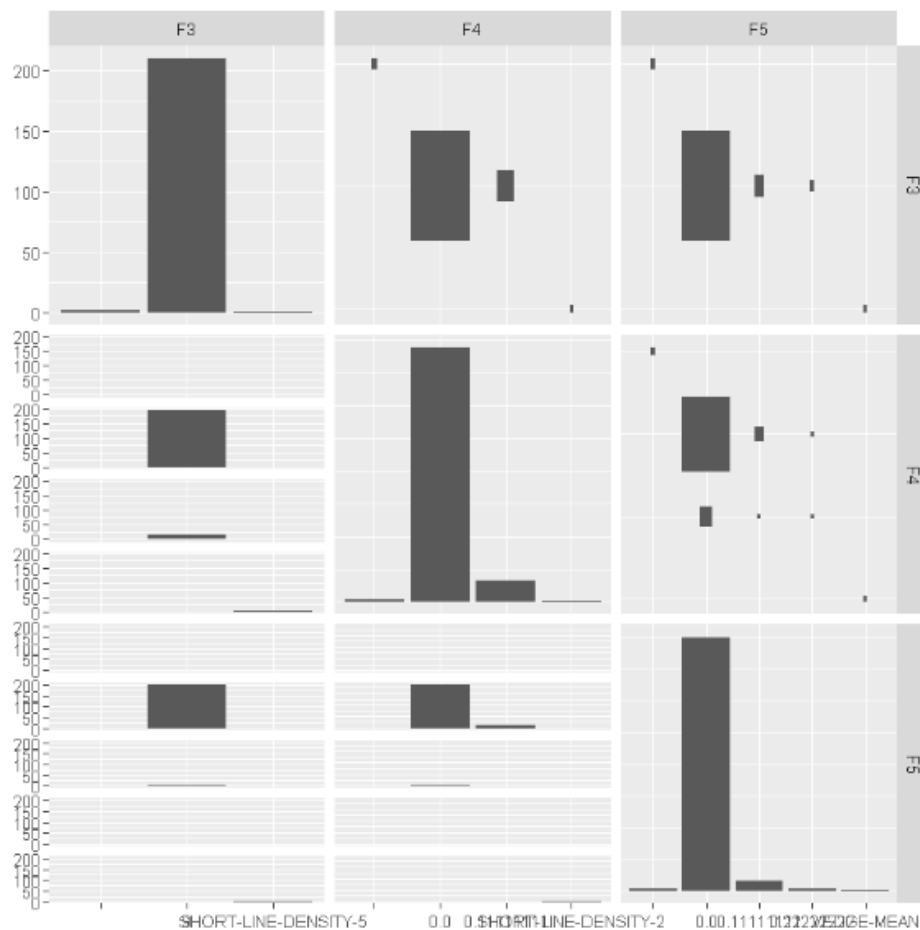
In [75]: colnames(segmentation) <- c("Class", paste0("F", 1:18))

In [76]: segmentation <- segmentation[, -1]

In [77]: remove_high_cardinality <- function(data, threshold) {
  high_cardinality_columns <- sapply(data, function(col) length(unique(col)))
  data[, !high_cardinality_columns]
}

In [78]: segmentation <- remove_high_cardinality(segmentation, threshold = 15)

In [79]: ggpairs(segmentation, upper = list(continuous = "points"), lower = list(conti
```



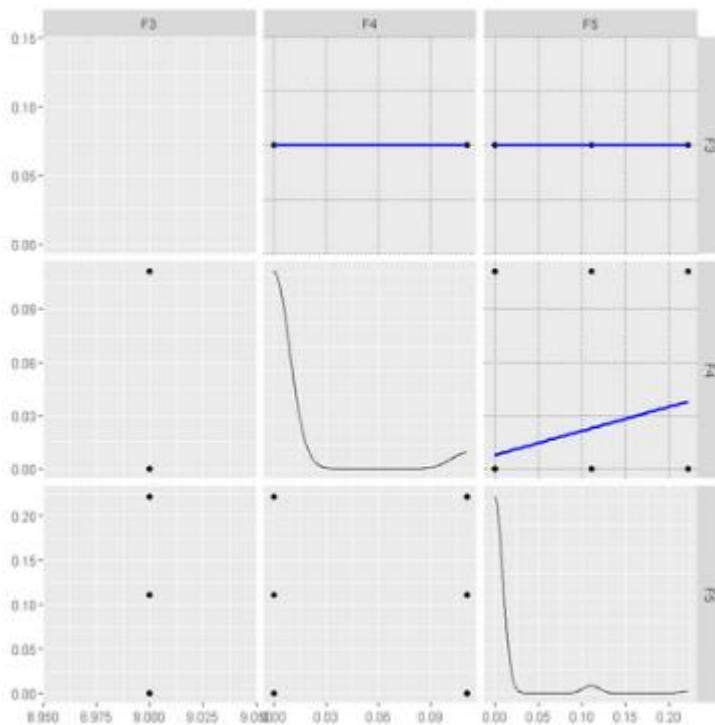
In [99]:

```
library(GGally)
library(ggplot2)

upper <- function(data, mapping, ...) {
  ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method = "lm", color = "blue", se = FALSE) +
    theme(panel.grid.major = element_line(color = "black", linetype = "dotted"),
          panel.grid.minor = element_blank())
}

ggpairs(segmentation,
        upper = list(continuous = upper))
```

Warning message:
 "Removed 3 rows containing non-finite values (`stat_density()`)."
 `geom_smooth()` using formula = 'y ~ x'
 Warning message:
 "Removed 3 rows containing non-finite values (`stat_smooth()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 `geom_smooth()` using formula = 'y ~ x'
 Warning message:
 "Removed 3 rows containing non-finite values (`stat_smooth()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 Warning message:
 "Removed 3 rows containing non-finite values (`stat_density()`)."
 `geom_smooth()` using formula = 'y ~ x'
 Warning message:
 "Removed 3 rows containing non-finite values (`stat_smooth()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 Warning message:
 "Removed 3 rows containing missing values (`geom_point()`)."
 Warning message:
 "Removed 3 rows containing non-finite values (`stat_density()`)."



In [98]:

```
library(GGally)
library(ggplot2)

lower <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_smooth() +
    theme_void() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}

ggpairs(segmentation,
        upper = list(continuous = lower))
```

Warning message:
 "Removed 3 rows containing non-finite values (`stat_density()`)."
 `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
 Warning message:
 "Removed 3 rows containing non-finite values (`stat_smooth()`)."
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "at -0.00055556"
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "radius 3.0864e-07"
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "all data on boundary of neighborhood. make span bigger"
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "pseudoinverse used at -0.00055556"
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "neighborhood radius 0.00055556"
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "reciprocal condition number 1"
 Warning message in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :
 "There are other near singularities as well. 0.012469"

In [97]:

```
library(GGally)
library(ggplot2)

upper <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method = "lm", color = "blue", se = FALSE) +
    theme(panel.grid.major = element_line(color = "grey", linetype = "dotted"),
          panel.grid.minor = element_blank())
}

diag <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_histogram() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}

ggpairs(segmentation,
        upper = list(continuous = upper),
        lower = list(continuous = "smooth"),
        diag = list(continuous = diag))
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:

"Removed 3 rows containing non-finite values (`stat_bin()`)."

`geom_smooth()` using formula = 'y ~ x'

Warning message:

"Removed 3 rows containing non-finite values (`stat_smooth()`)."

Warning message:

"Removed 3 rows containing missing values (`geom_point()`)."

`geom_smooth()` using formula = 'y ~ x'

Warning message:

"Removed 3 rows containing non-finite values (`stat_smooth()`)."

Warning message:

"Removed 3 rows containing missing values (`geom_point()`)."

Warning message:

"Removed 3 rows containing non-finite values (`stat_smooth()`)."

Warning message:

"Removed 3 rows containing missing values (`geom_point()`)."

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:

"Removed 3 rows containing non-finite values (`stat_bin()`)."

`geom_smooth()` using formula = 'y ~ x'

Warning message:

"Removed 3 rows containing non-finite values (`stat_smooth()`)."

Warning message:

"Removed 3 rows containing missing values (`geom_point()`)."

Warning message:

"Removed 3 rows containing non-finite values (`stat_smooth()`)."

Warning message:

"Removed 3 rows containing missing values (`geom_point()`)."

Warning message:

"Removed 3 rows containing non-finite values (`stat_smooth()`)."

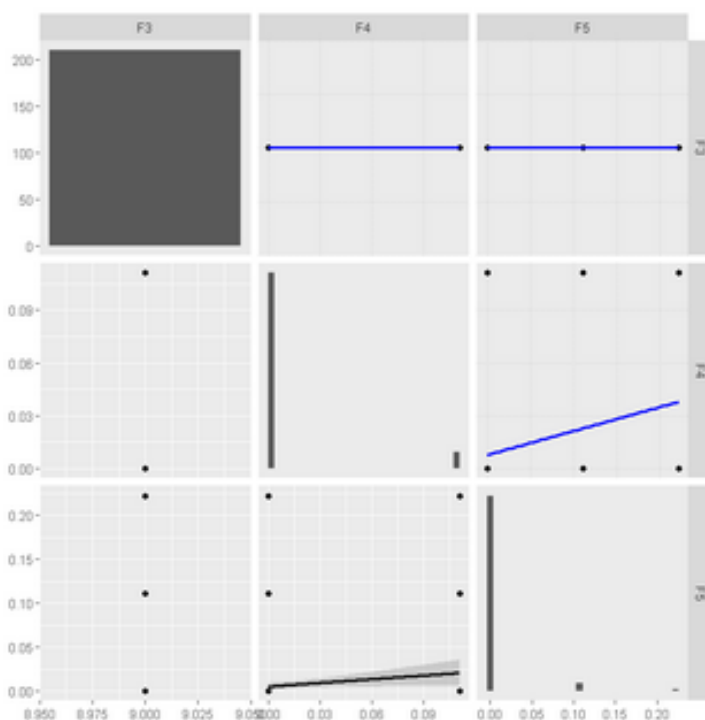
Warning message:

"Removed 3 rows containing missing values (`geom_point()`)."

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning message:

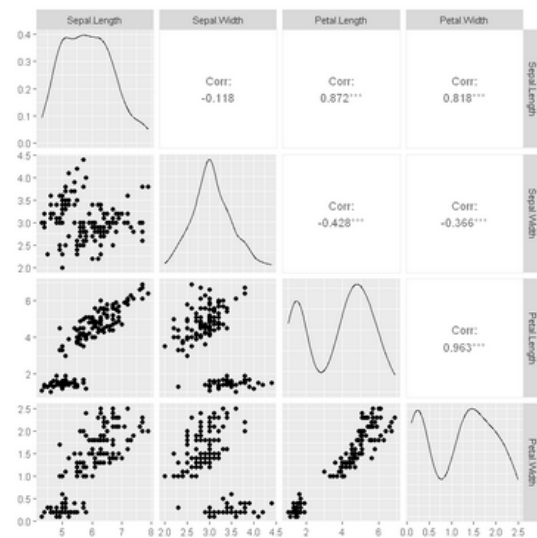
"Removed 3 rows containing non-finite values (`stat_bin()`)."



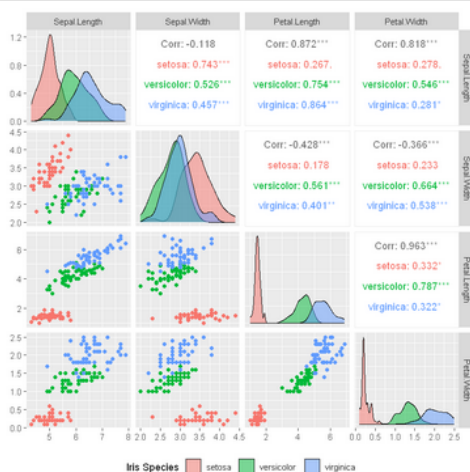
3. Draw a graph matrix for IRIS Dataset.

- Use customized functions for the panels to scatterplot matrix with diagonal panels showing the density plot of each variable. One half of the scatterplot matrix shows the scatterplots for each pair of variables while the other half shows the corresponding Pearson correlation coefficient of each pair.
- Show the data from each species separately, or in a different color on the same plot.

In [101]: `ggpairs(iris[,1:4])`



In [102]: `ggpairs(iris, columns = c(1:4), aes(color = Species), legend = 1, diag = list(continuous = wrap("densityDiag", alpha=0.5))) + theme(legend.position = "bottom") + labs(fill = "Iris Species")`



In [103]: `upper <- function(data, mapping, ...){
 ggplot(data = data, mapping = mapping) +
 geom_point() +
 geom_smooth(method = "lm", color = "Species", se = FALSE) +
 theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}`

In [104]: `diag <- function(data, mapping, ...){
 ggplot(data = data, mapping = mapping) +
 geom_density(fill = "skyblue", color = "Species") +
 theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}`

```
In [103]: upper <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method = "lm", color = "Species", se = FALSE) +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}
```

```
In [104]: diag <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_density(fill = "skyblue", color = "Species") +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}
```

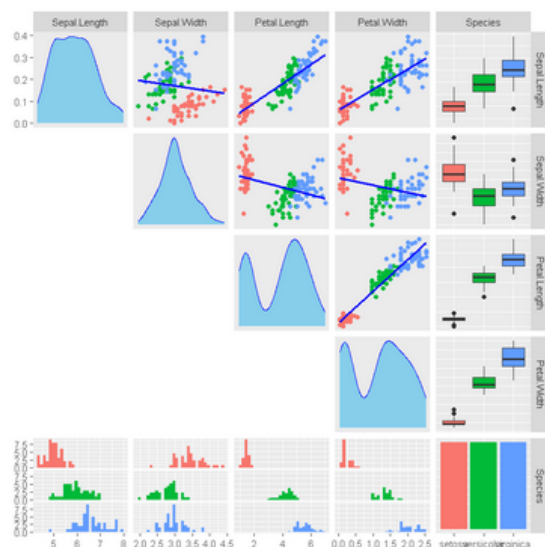
```
In [117]: library(GGally)
library(ggplot2)

upper <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method = "lm", color = "blue", se = FALSE) +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}

diag <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_density(fill = "skyblue", color = "blue") +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
}

ggpairs(iris,
  upper = list(continuous = upper),
  lower = list(continuous = "blank"),
  diag = list(continuous = diag),
  mapping = aes(color = Species))

`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
In [125]: lower_cor <- function(data, mapping, ...) {
  num_data <- data[, sapply(data, is.numeric)]

  corr <- cor(num_data, use = "complete.obs")

  corr_lower <- lower.tri(corr)

  row_idx <- row(corr)[corr_lower]
  col_idx <- col(corr)[corr_lower]

  cor_df <- data.frame(x = col_idx, y = row_idx, label = round(corr[corr_lower], 2))

  ggplot(data = cor_df, mapping = aes(x = x, y = y, label = label)) +
    geom_text(size = 6) +
    theme_void() +
    theme(legend.position = "none")
}
```



```
In [125]: lower_cor <- function(data, mapping, ...) {
  num_data <- data[, sapply(data, is.numeric)]
  corr <- cor(num_data, use = "complete.obs")
  corr_lower <- lower.tri(corr)

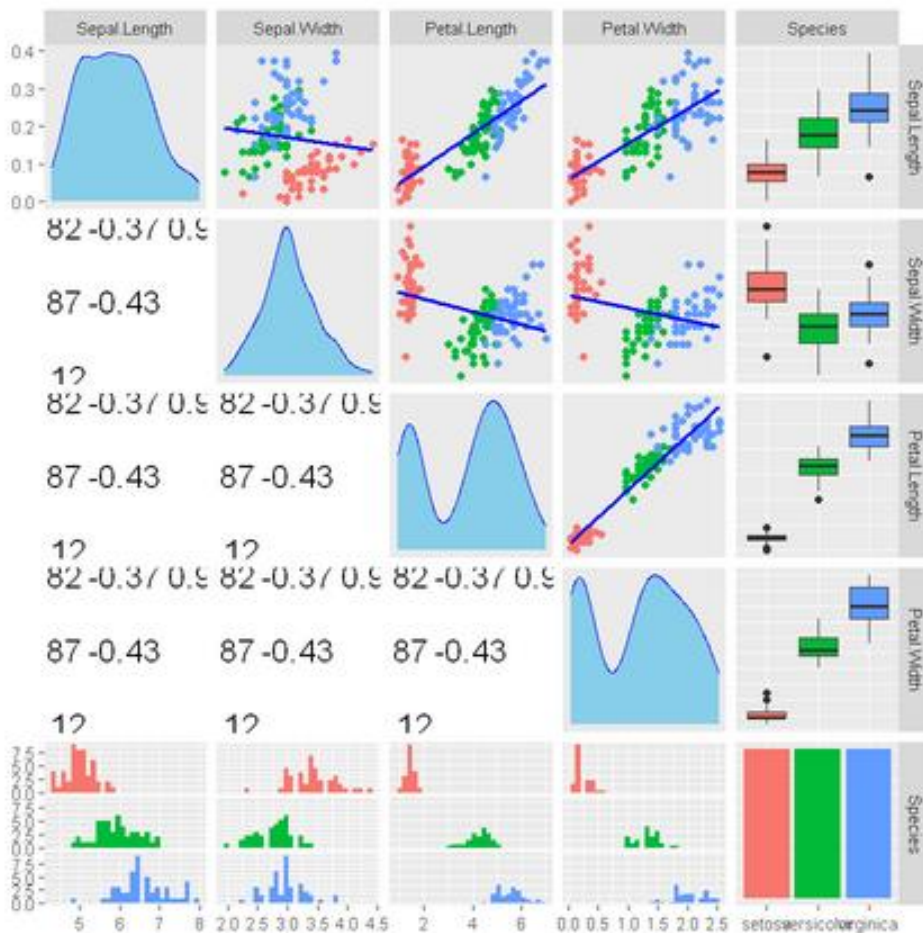
  row_idx <- row(corr)[corr_lower]
  col_idx <- col(corr)[corr_lower]

  cor_df <- data.frame(x = col_idx, y = row_idx, label = round(corr[corr_lower], 2))

  ggplot(data = cor_df, mapping = aes(x = x, y = y, label = label)) +
    geom_text(size = 6) +
    theme_void() +
    theme(legend.position = "none")
}
```

```
In [126]: ggpairs(iris,
  upper = list(continuous = upper),
  lower = list(continuous = lower_cor),
  diag = list(continuous = diag),
  mapping = aes(color = Species))

`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



4. Draw a graph matrix for mushroom dataset (available in UCI repository)

• Use customized functions for the panels to Scatter plot with correlations in the upper triangle,

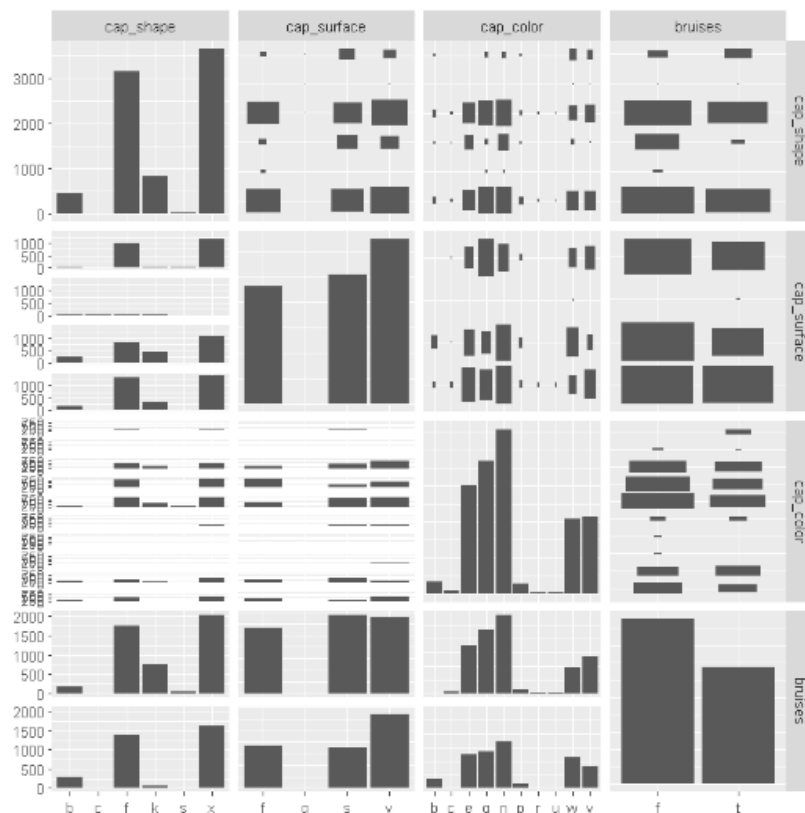
smoothing lines in the lower triangle, and histograms on the diagonal.

```
In [130]: library(GGally)
library(ggplot2)

mushroom <- read.csv("C:/Users/gyanada/Downloads/mushroom/agaricus-lepiota.data", header = FALSE)
```

```
In [131]: colnames(mushroom) <- c("class", "cap_shape", "cap_surface", "cap_color",
    "bruises", "odor", "gill_attachment", "gill_spacing",
    "gill_size", "gill_color", "stalk_shape", "stalk_root",
    "stalk_surface_above_ring", "stalk_surface_below_ring",
    "stalk_color_above_ring", "stalk_color_below_ring",
    "veil_type", "veil_color", "ring_number", "ring_type",
    "spore_print_color", "population", "habitat")
```

```
In [132]: ggpairs(mushroom,
    columns = c(2:5),
    upper = list(continuous = wrap("cor", method = "pearson")),
    diag = list(continuous = wrap("densityDiag", alpha = 0.5)))
```



```
In [1]: library(GGally)

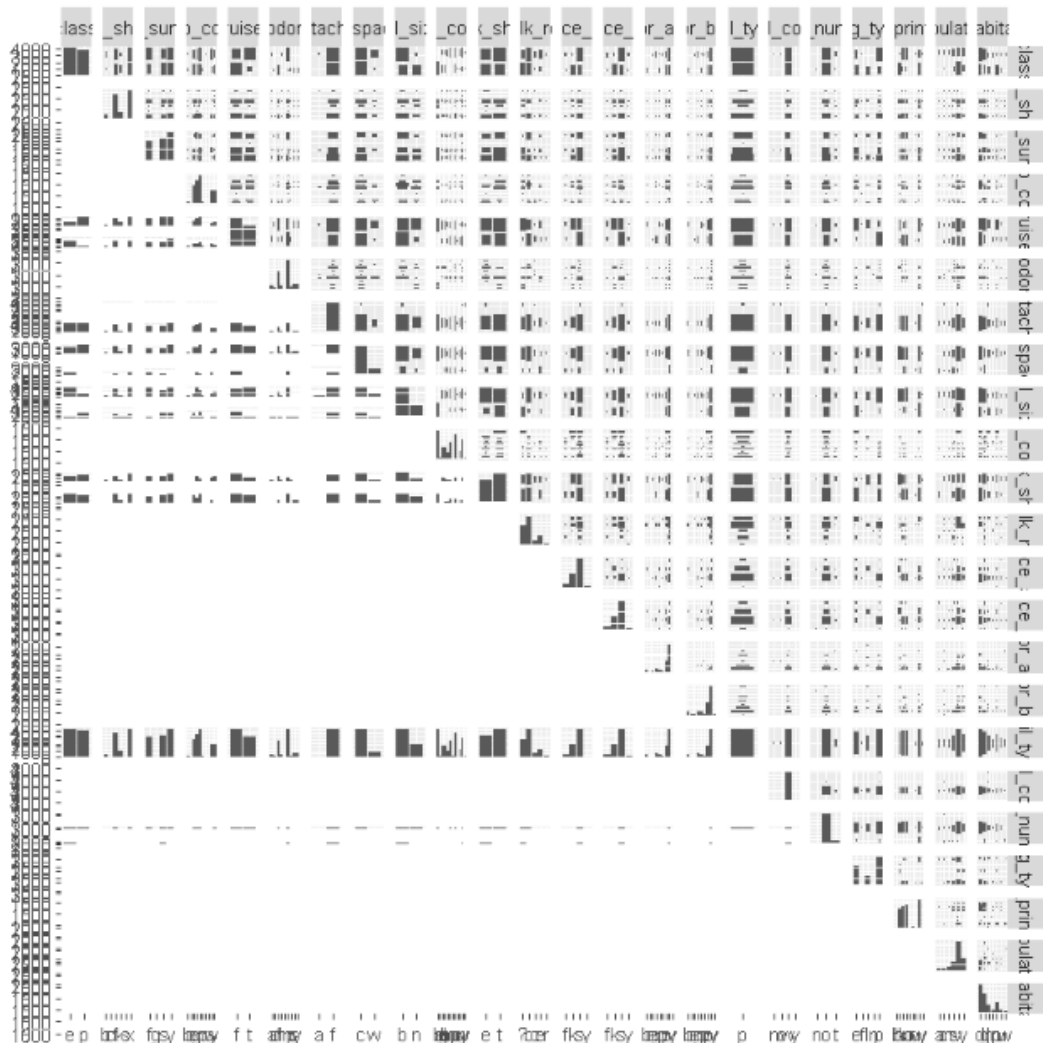
mushroom <- read.csv("C:/Users/gyanada/Downloads/mushroom/agaricus-lepiota.data", header = FALSE)

colnames(mushroom) <- c("class", "cap_shape", "cap_surface", "cap_color",
  "bruises", "odor", "gill_attachment", "gill_spacing",
  "gill_size", "gill_color", "stalk_shape", "stalk_root",
  "stalk_surface_above_ring", "stalk_surface_below_ring",
  "stalk_color_above_ring", "stalk_color_below_ring",
  "veil_type", "veil_color", "ring_number", "ring_type",
  "spore_print_color", "population", "habitat")

ggpairs(mushroom)
```

Loading required package: ggplot2

Warning message:
 "package 'ggplot2' was built under R version 4.2.3"
 Registered S3 method overwritten by 'GGally':
 method from
 +.gg ggplot2



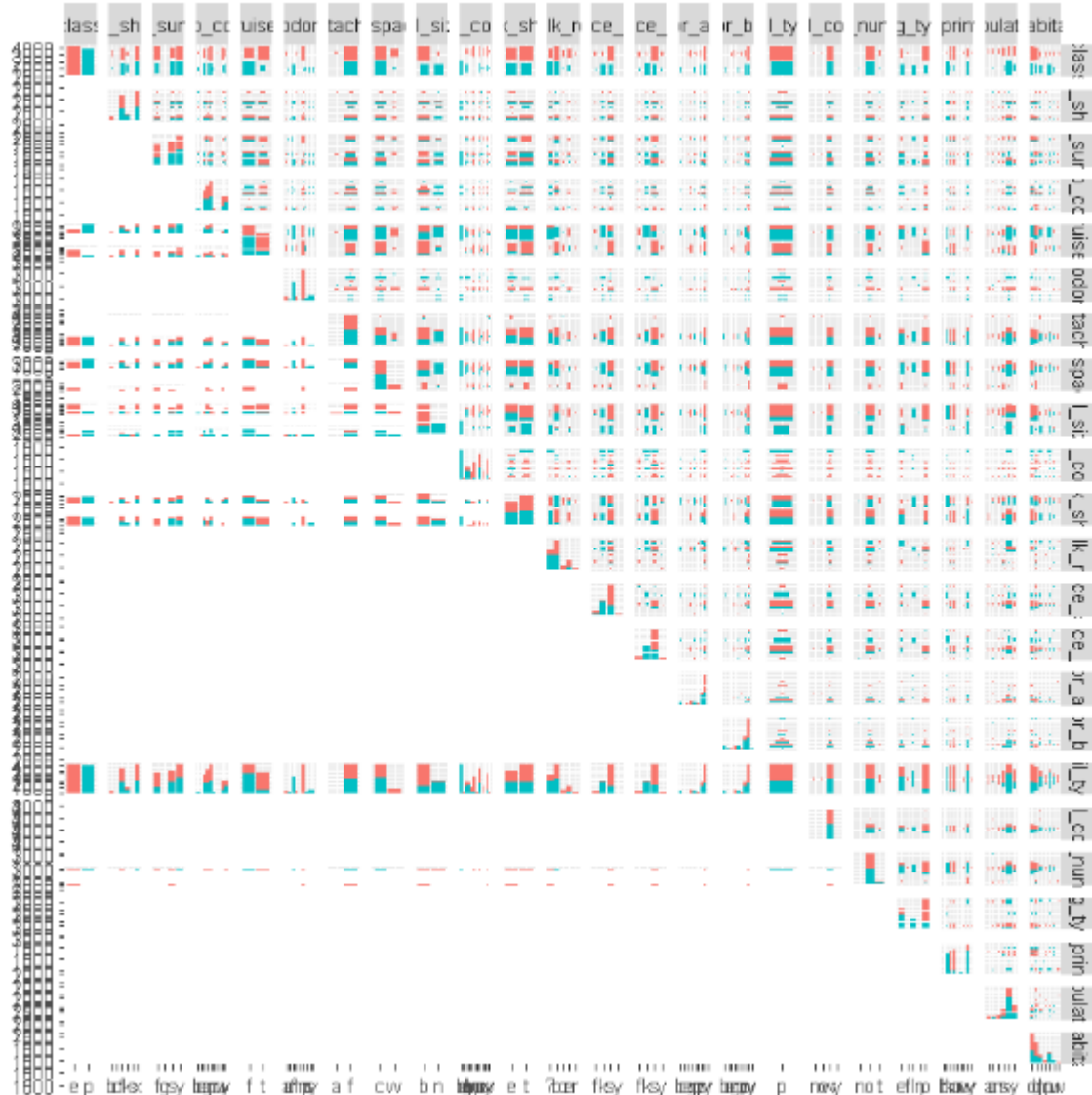
In [2]:

```
library(GGally)

mushroom <- read.csv("C:/Users/gyanada/Downloads/mushroom/agaricus-lepiota.data", header = FALSE)

colnames(mushroom) <- c("class", "cap_shape", "cap_surface", "cap_color",
  "bruises", "odor", "gill_attachment", "gill_spacing",
  "gill_size", "gill_color", "stalk_shape", "stalk_root",
  "stalk_surface_above_ring", "stalk_surface_below_ring",
  "stalk_color_above_ring", "stalk_color_below_ring",
  "veil_type", "veil_color", "ring_number", "ring_type",
  "spore_print_color", "population", "habitat")

ggpairs(mushroom,
  upper = list(continuous = wrap("cor", method = "pearson")),
  lower = list(continuous = wrap("smooth", method = "loess")),
  diag = list(continuous = wrap("densityDiag", alpha = 0.5)),
  mapping = aes(color = class))
```





```
In [11]: library(Ggally)

upper_fun <- function(data, mapping, ...) {
  ggally_points(data = data, mapping = mapping, ...) +
    geom_smooth(method = "lm", color = "blue", se = FALSE)
}

lower_fun <- function(data, mapping, ...) {
  ggally_smooth(data = data, mapping = mapping, method = "loess", ...) +
    geom_smooth(method = "loess", color = "red", se = FALSE)
}

diag_fun <- function(data, mapping, ...) {
  ggally_density(data = data, mapping = mapping, ...)
}

graph_matrix <- ggpairs(mushroom,
  upper = list(continuous = upper_fun),
  lower = list(continuous = lower_fun),
  diag = list(continuous = diag_fun))

print(graph_matrix)
```

