

lattice

Katarzyna Tokarczuk

2024-12-28

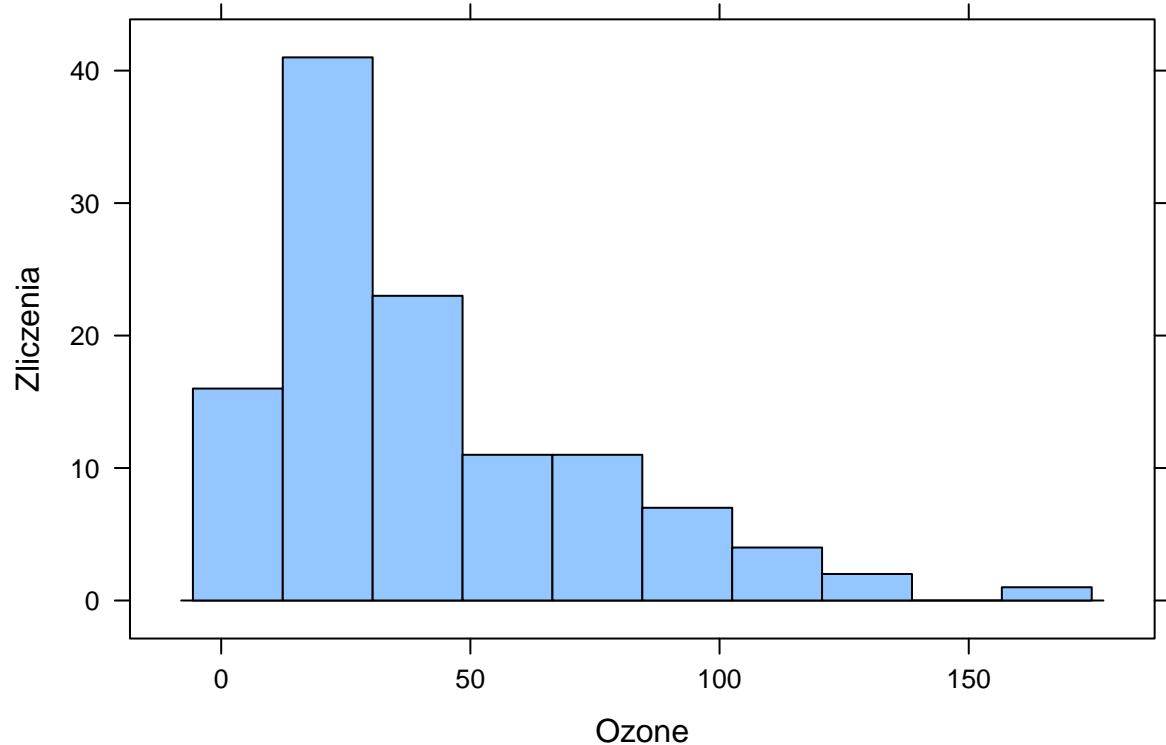
```
library("plotly")
library('ggplot2')
library('tidyverse')
library("lattice")
library("latticeExtra")
```

Example 1

```
#a)
str(airquality)

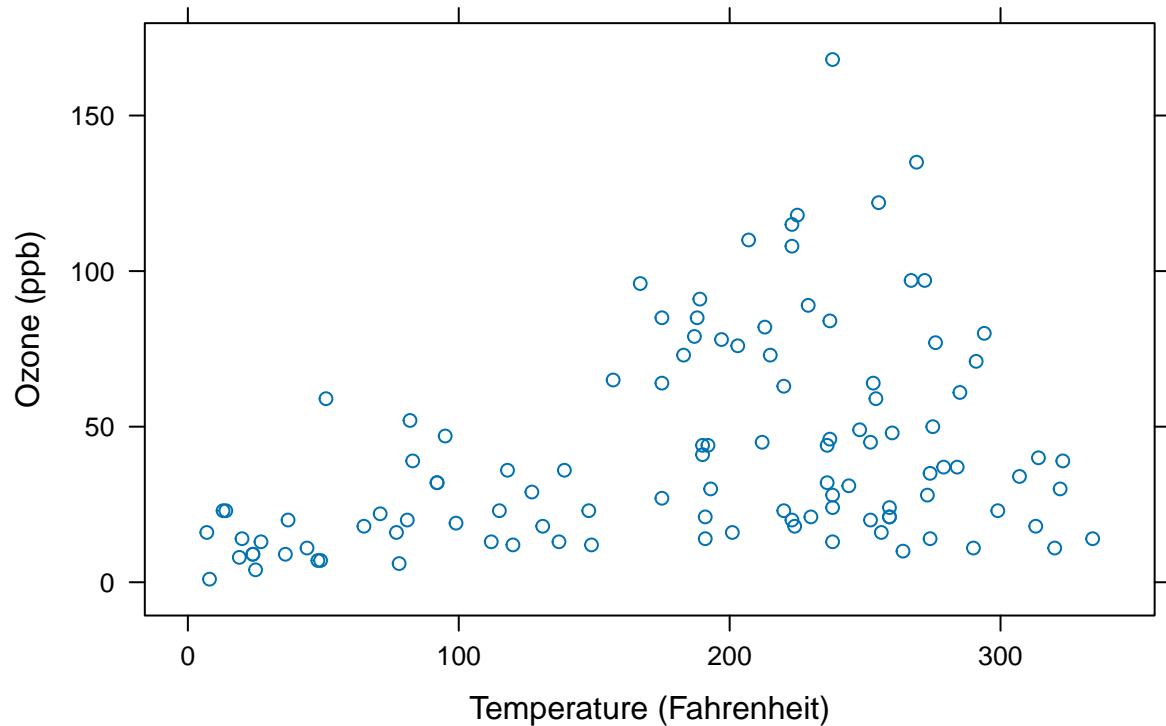
## 'data.frame':   153 obs. of  6 variables:
## $ Ozone    : int  41 36 12 18 NA 28 23 19 8 NA ...
## $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
## $ Wind     : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
## $ Temp     : int  67 72 74 62 56 66 65 59 61 69 ...
## $ Month    : int  5 5 5 5 5 5 5 5 5 5 ...
## $ Day      : int  1 2 3 4 5 6 7 8 9 10 ...

histogram(~ Ozone, data = airquality,
           nint=10,
           type="count")
```

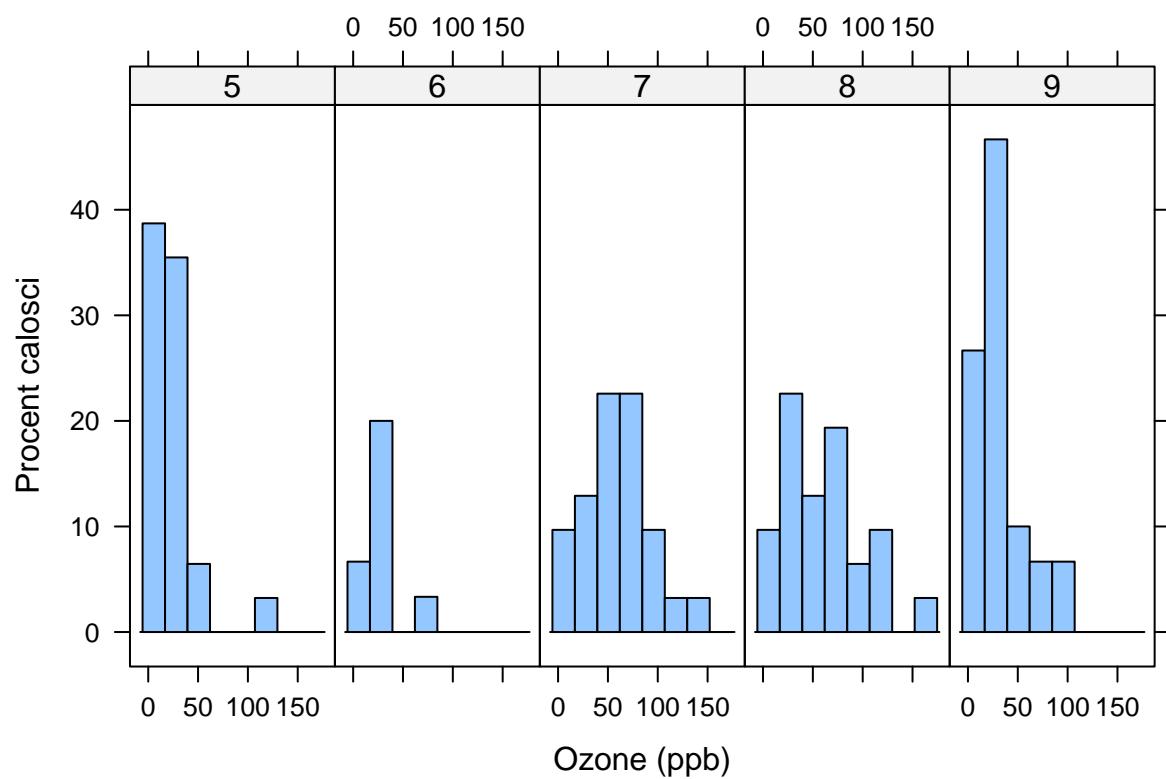


```
#b)
xyplot(Ozone ~ Solar.R, data = airquality,
       main = "Air Quality",
       xlab = "Temperature (Fahrenheit)",
       ylab = "Ozone (ppb")
)
```

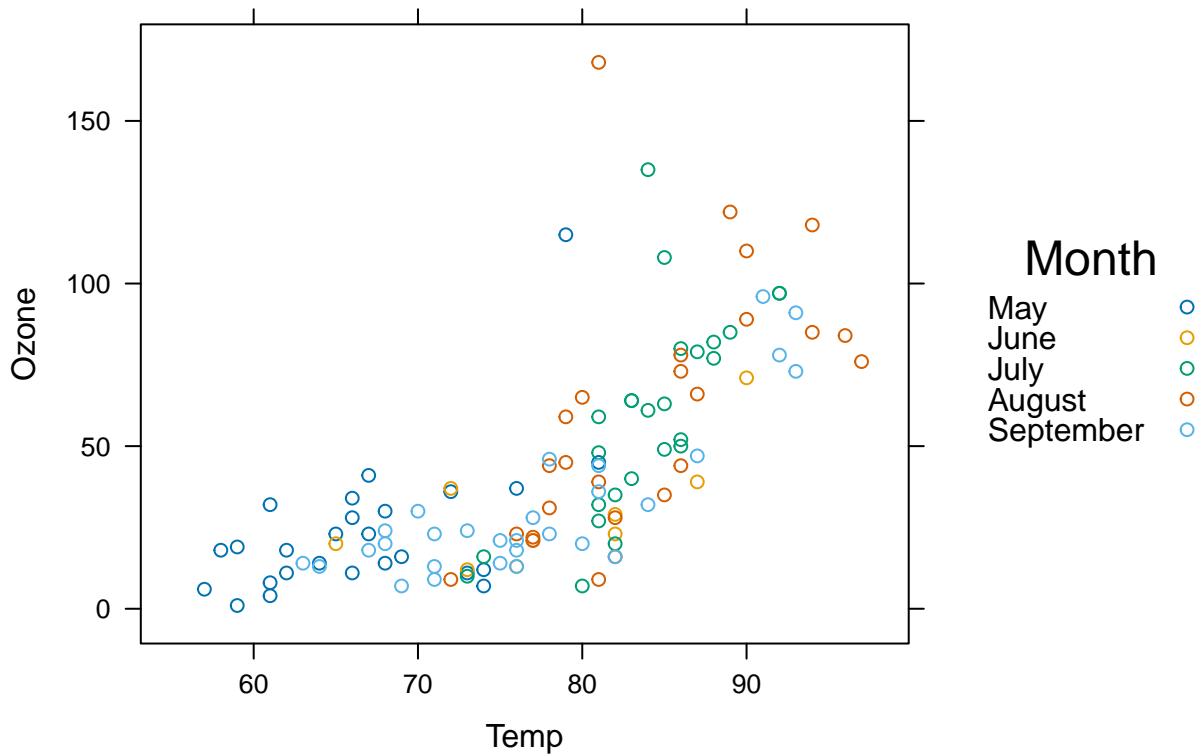
Air Quality



```
#c)
histogram(~Ozone|factor(Month),
           data = airquality,
           layout=c(5,1),
           xlab="Ozone (ppb)")
```



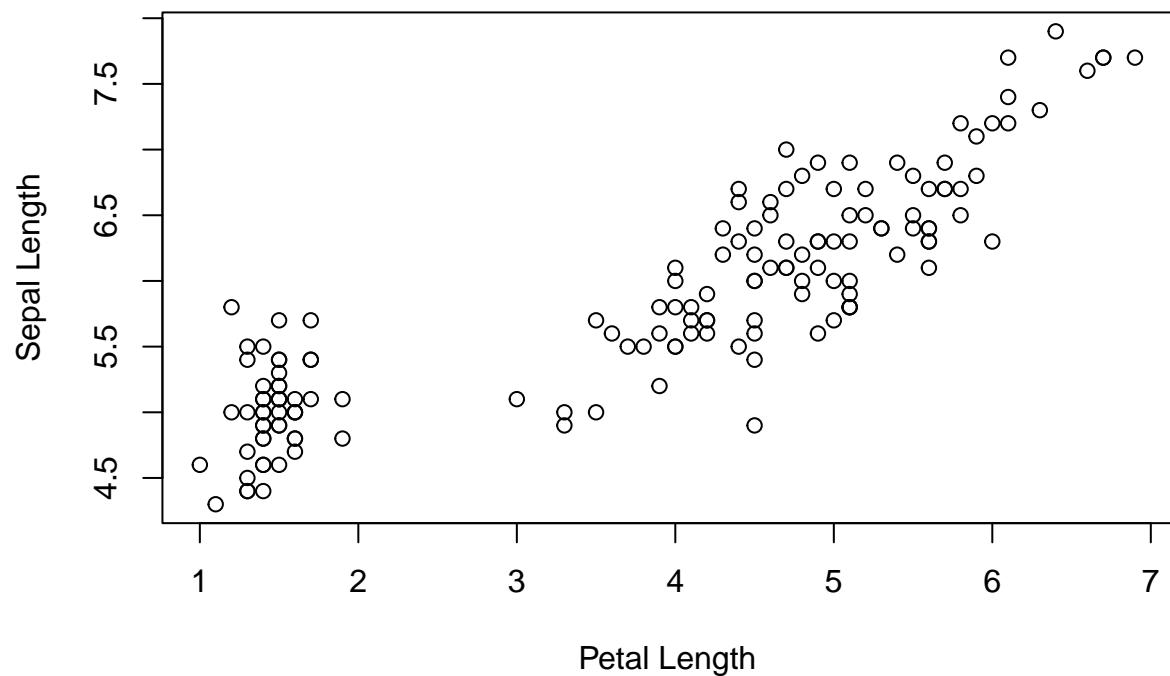
```
#d)
xyplot(Ozone ~ Temp, airquality, groups = Month,
       auto.key = list(space = "right",
                      title = "Month",
                      text = month.name[5:9]))
```



Task 1

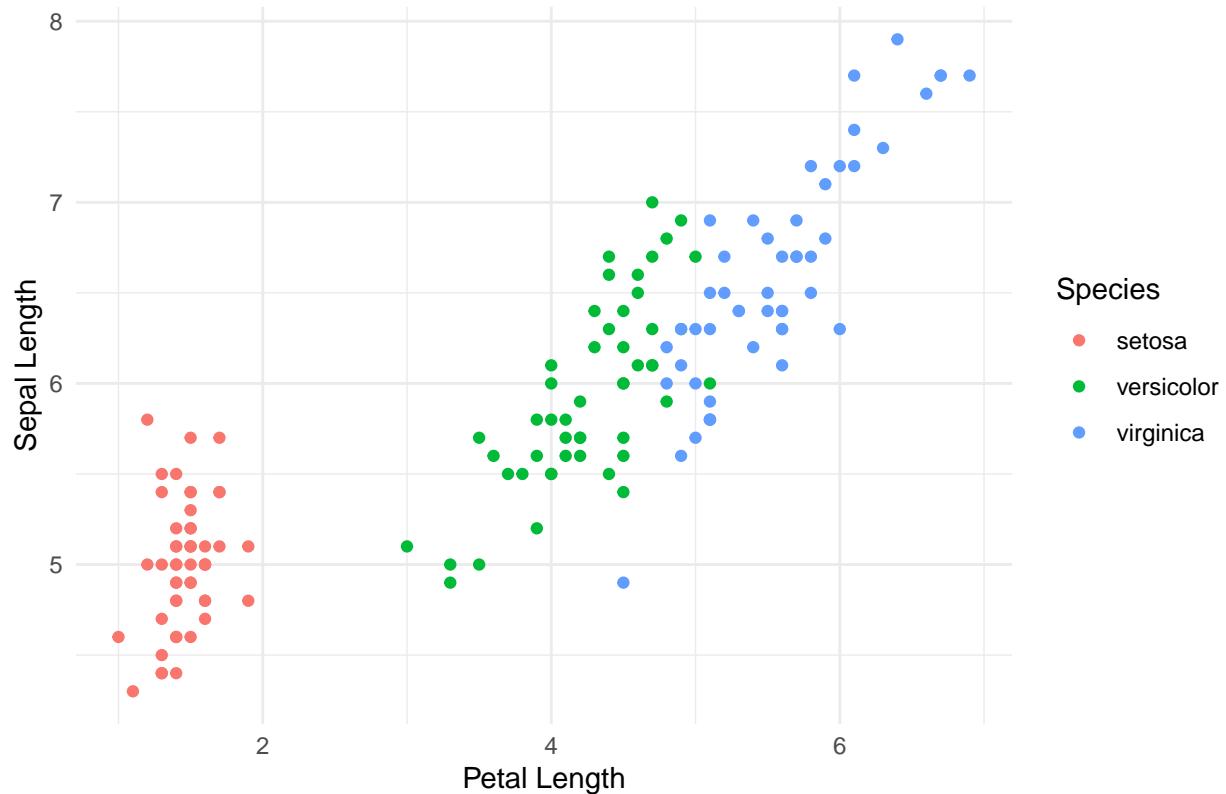
```
#a
plot(iris$Petal.Length, iris$Sepal.Length,
      main = "Iris scatterplot",
      xlab = "Petal Length",
      ylab = "Sepal Length")
```

Iris scatterplot



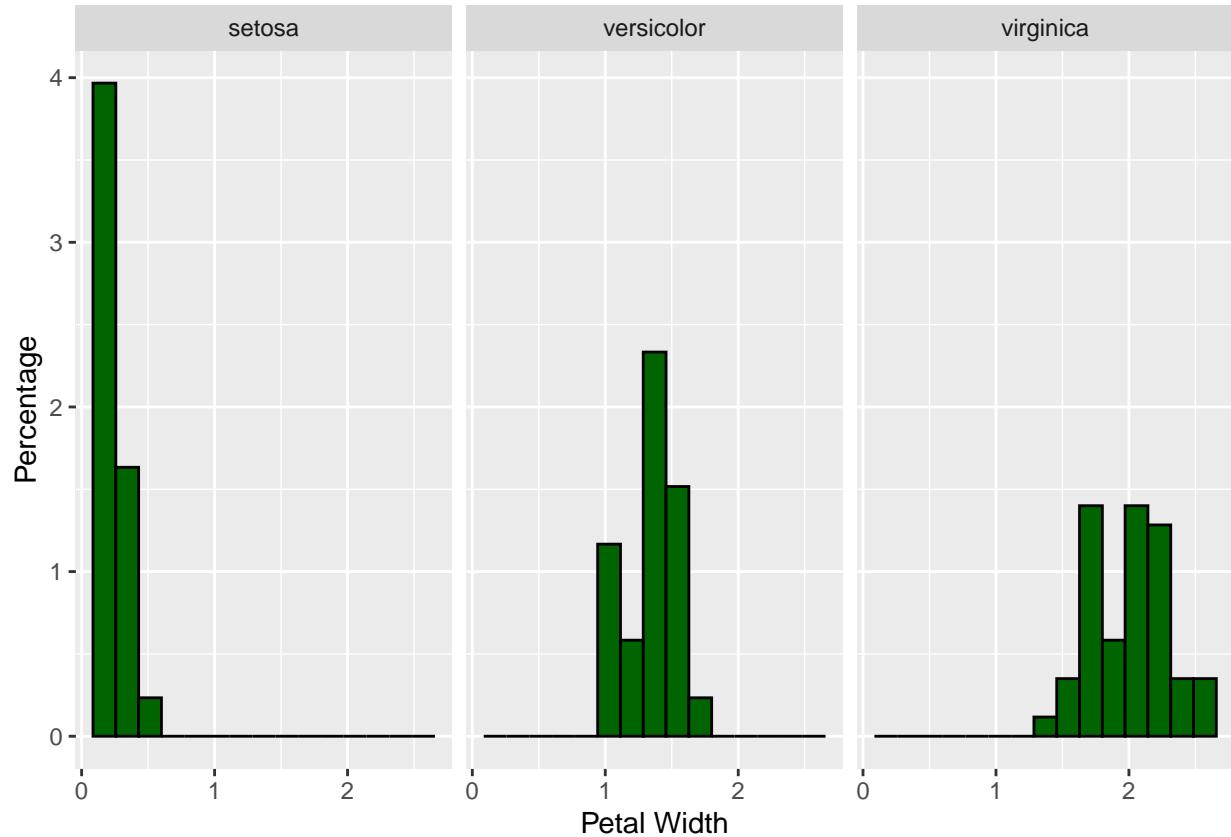
```
#b  
ggplot(iris, aes(x = Petal.Length, y = Sepal.Length, color = Species)) +  
  geom_point() +  
  labs(title = "Iris scatterplot", x = "Petal Length", y = "Sepal Length") +  
  theme_minimal()
```

Iris scatterplot

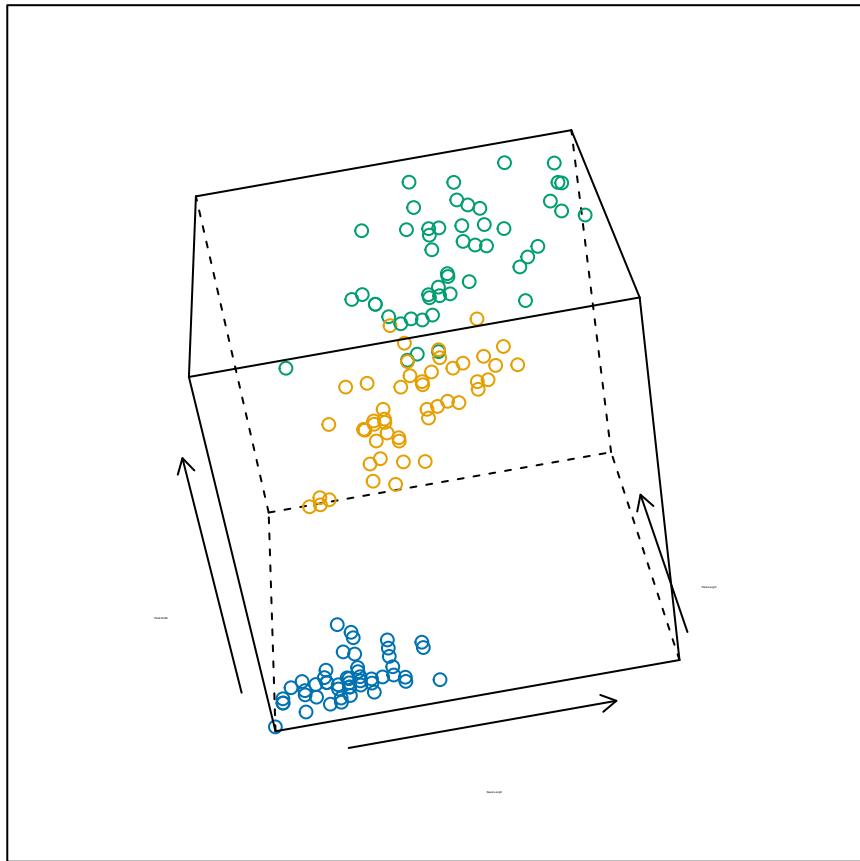


```
#c
ggplot(iris, aes(x = Petal.Width)) +
  geom_histogram(bins = 15, fill = "darkgreen", color = "black", aes(y = ..density..)) +
  facet_wrap(~ Species, nrow = 1) +
  labs(y = "Percentage", x = "Petal Width")

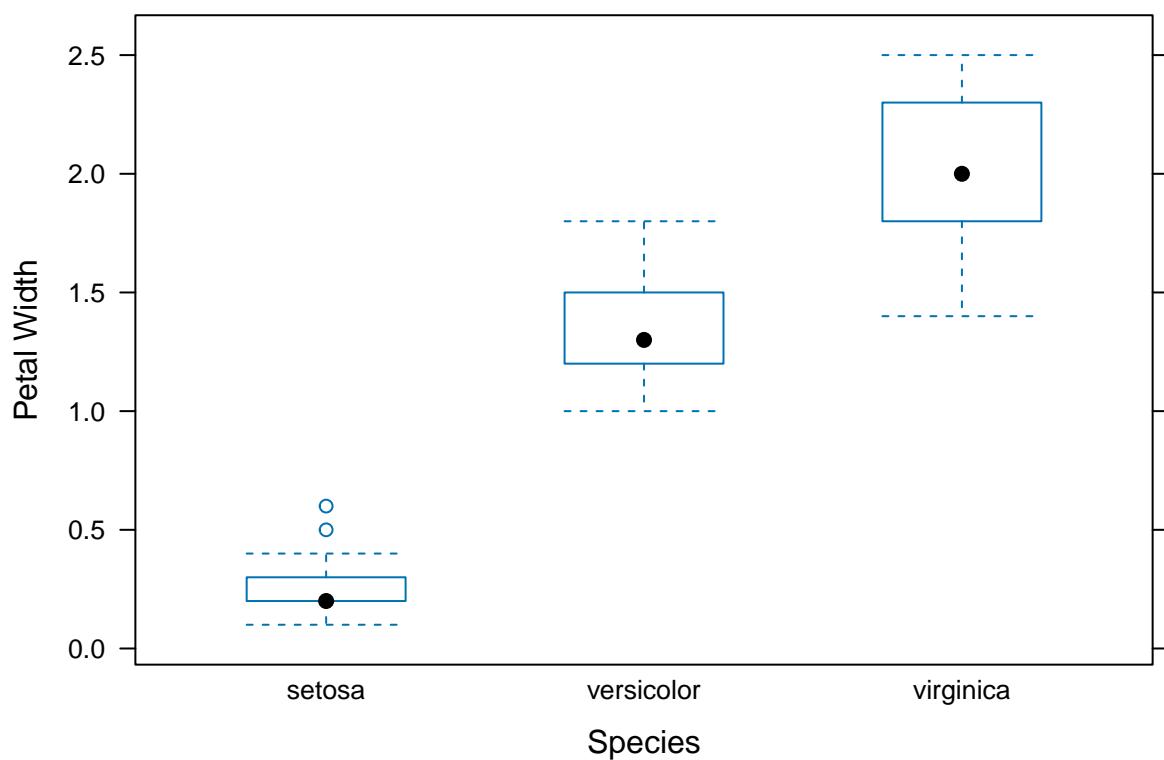
## Warning: The dot-dot notation ('..density..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(density)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



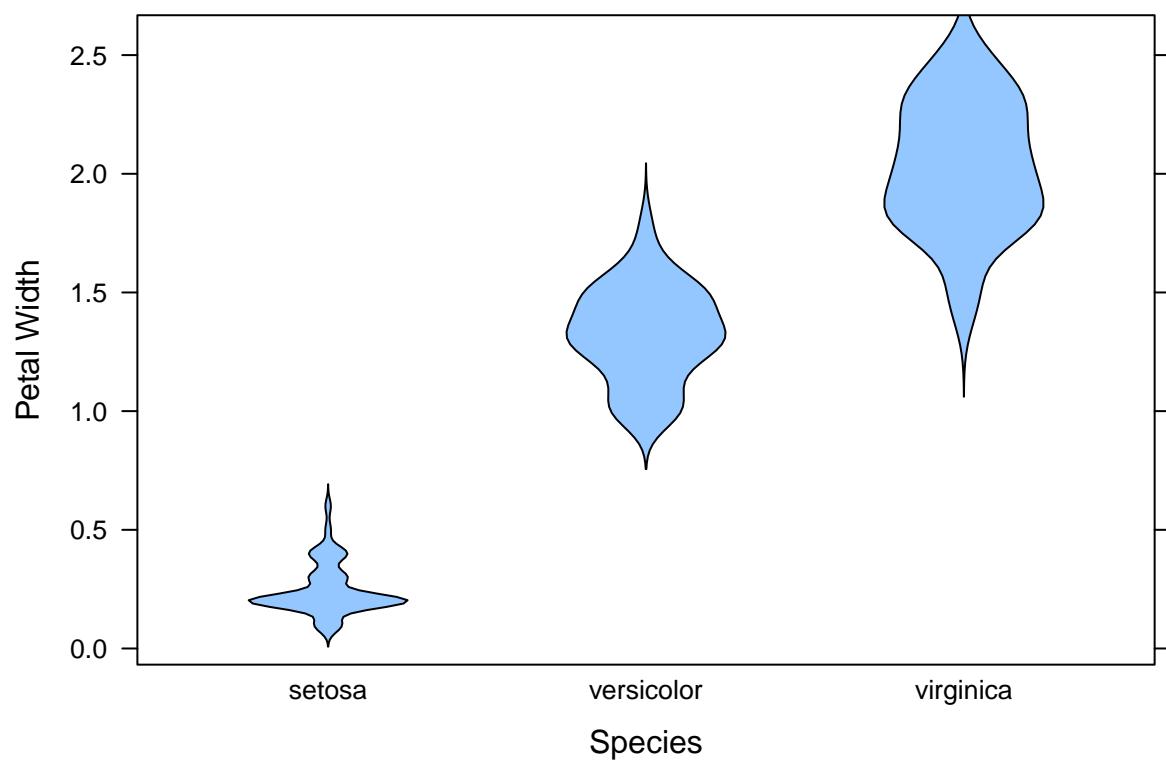
```
#d
cloud(Petal.Width ~ Sepal.Length * Petal.Length, data = iris, groups = Species,
      screen = list(x = -60, y = 0, z = 10),
      par.settings = list(fontsize = list(text = 0.8)))
```



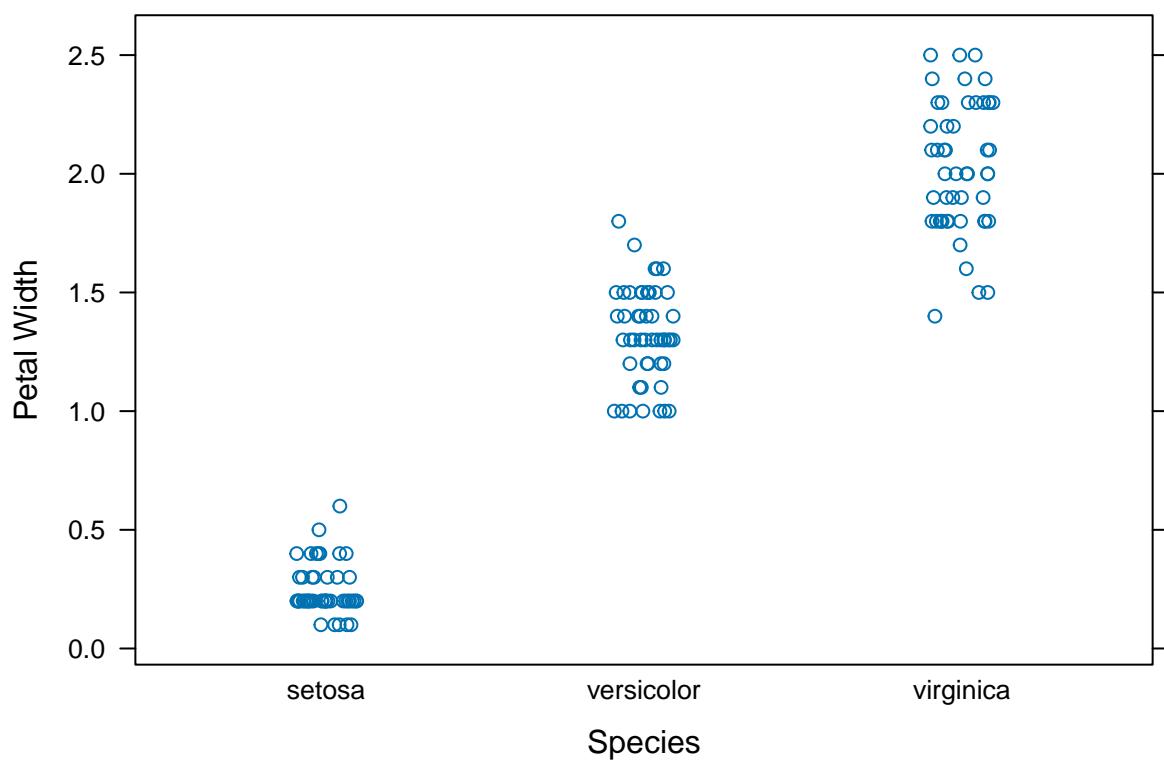
```
#e  
bwplot(Petal.Width ~ Species, data = iris, xlab = "Species", ylab = "Petal Width")
```



```
bwplot(Petal.Width ~ Species, data = iris,  
       panel = panel.violin, xlab = "Species", ylab = "Petal Width")
```

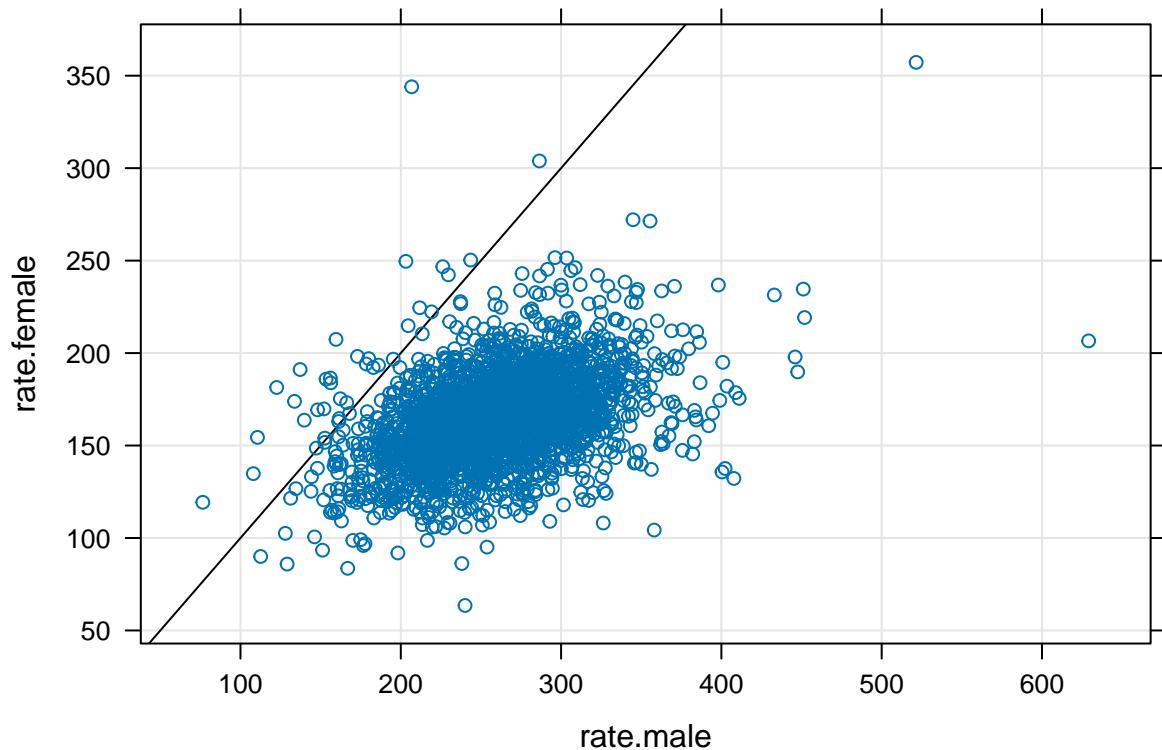


```
#f  
stripplot(Petal.Width ~ Species, data = iris, jitter.data = TRUE,  
         xlab = "Species", ylab = "Petal Width")
```

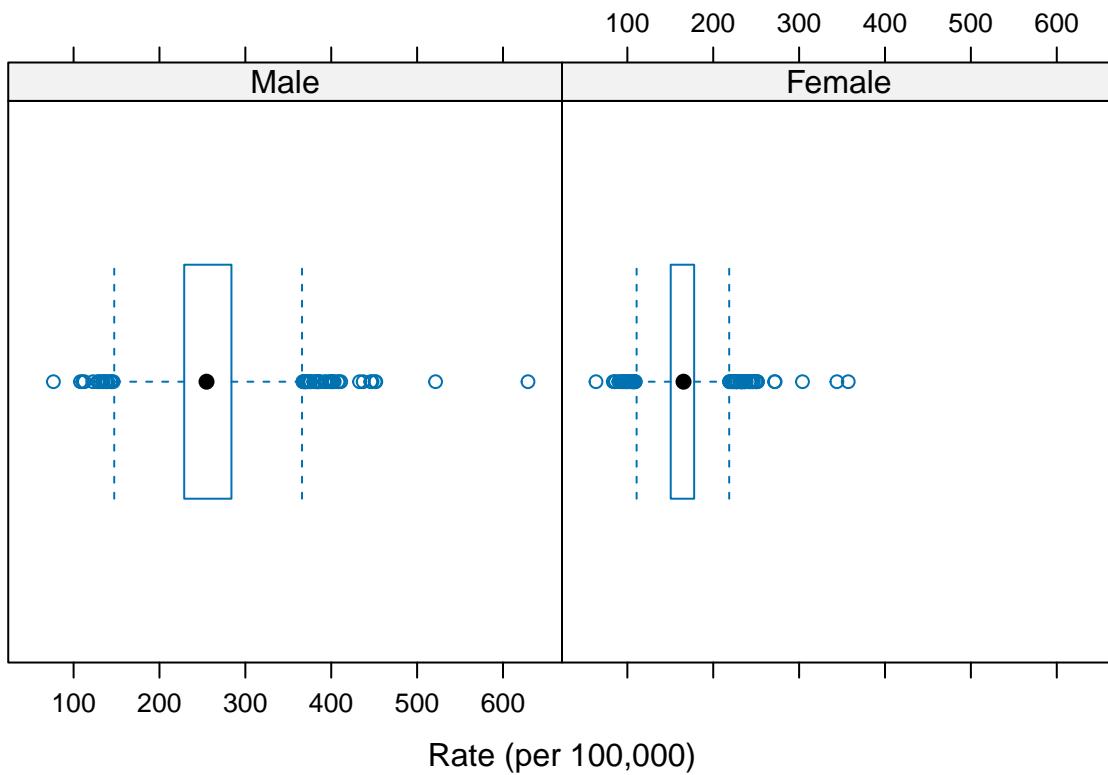


Example 2

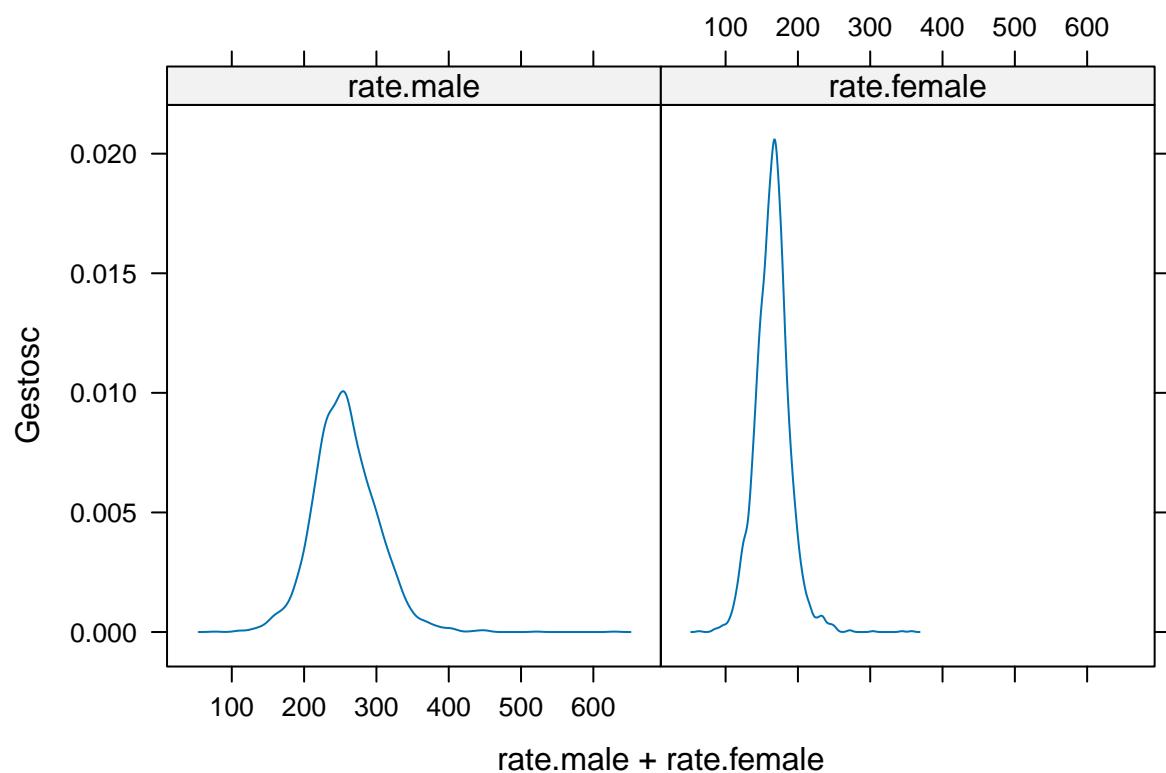
```
#a
xyplot(rate.female ~ rate.male,
       data = USCancerRates,
       grid = TRUE, abline = c(0, 1),
       as.table=TRUE)
```



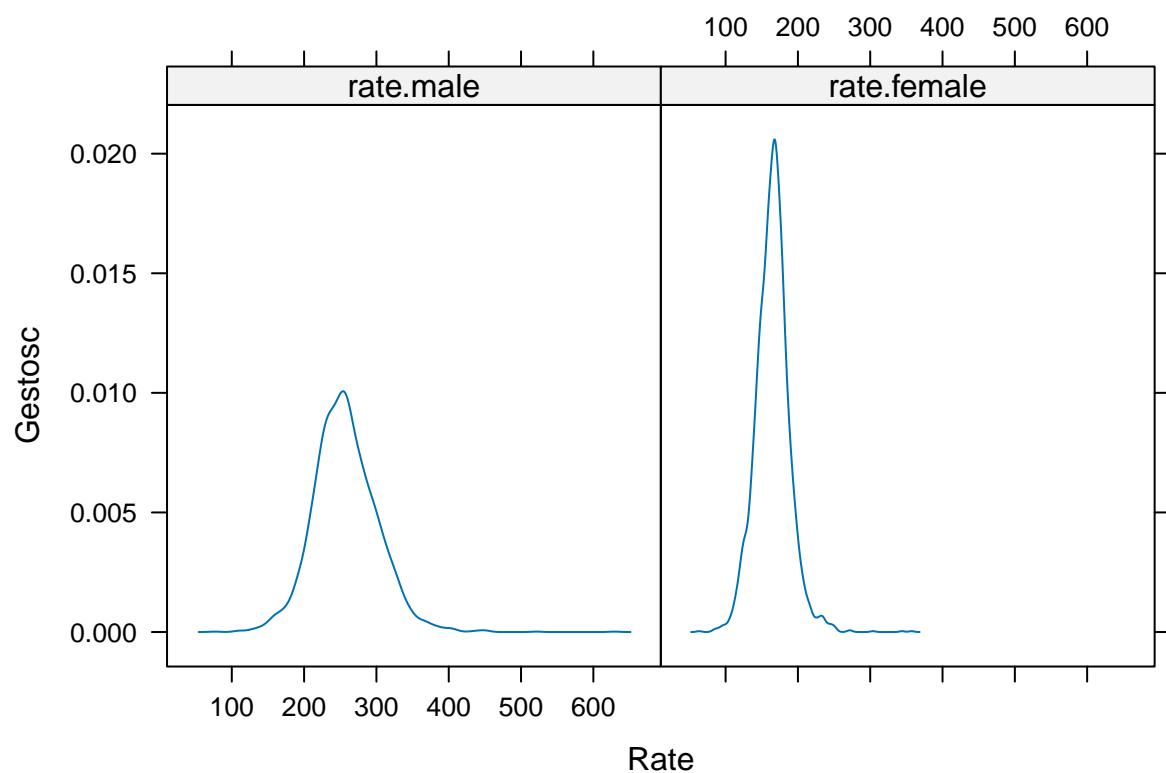
```
#b  
bwplot(~ rate.male + rate.female,  
      data = USCancerRates,  
      outer = TRUE,  
      xlab="Rate (per 100,000)",  
      strip = strip.custom(factor.levels = c("Male", "Female"))  
)
```



```
#c
dens_plot <-
  densityplot(~ rate.male + rate.female,
              data = USCancerRates, outer = TRUE,
              plot.points = FALSE, as.table = TRUE)
print(dens_plot)
```



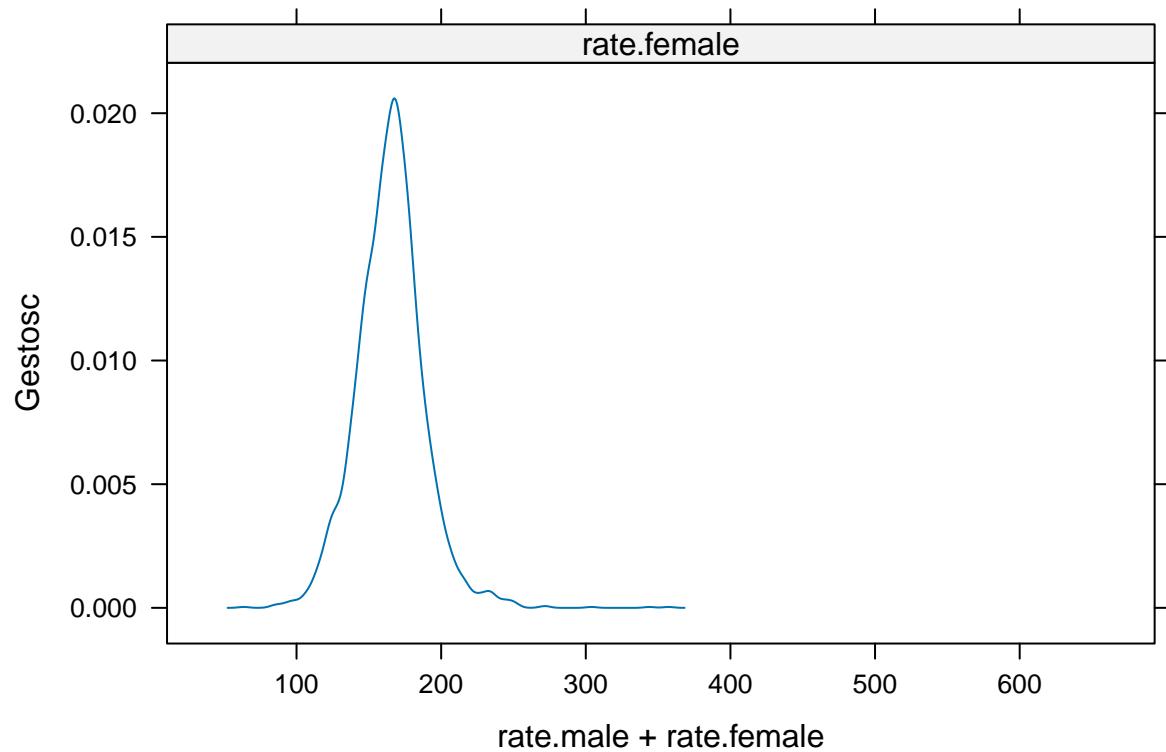
```
update(dens_plot, xlab = "Rate")
```



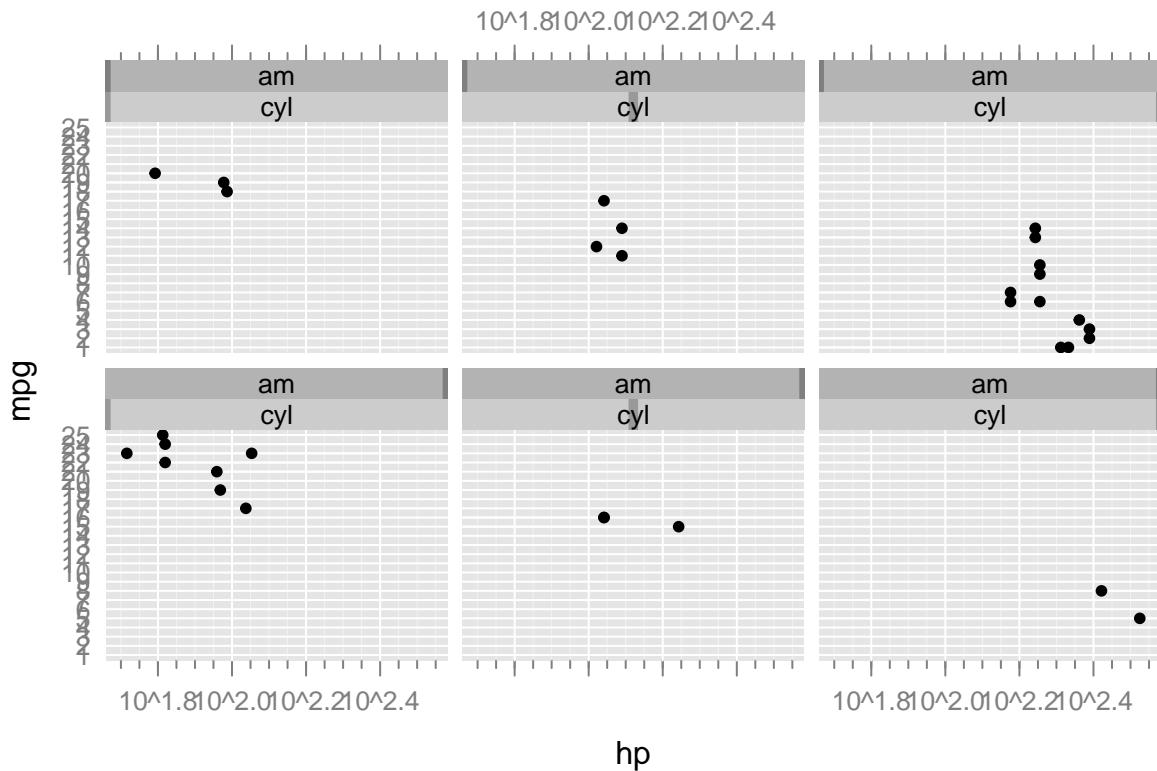
```
dim(dens_plot)
```

```
## [1] 2
```

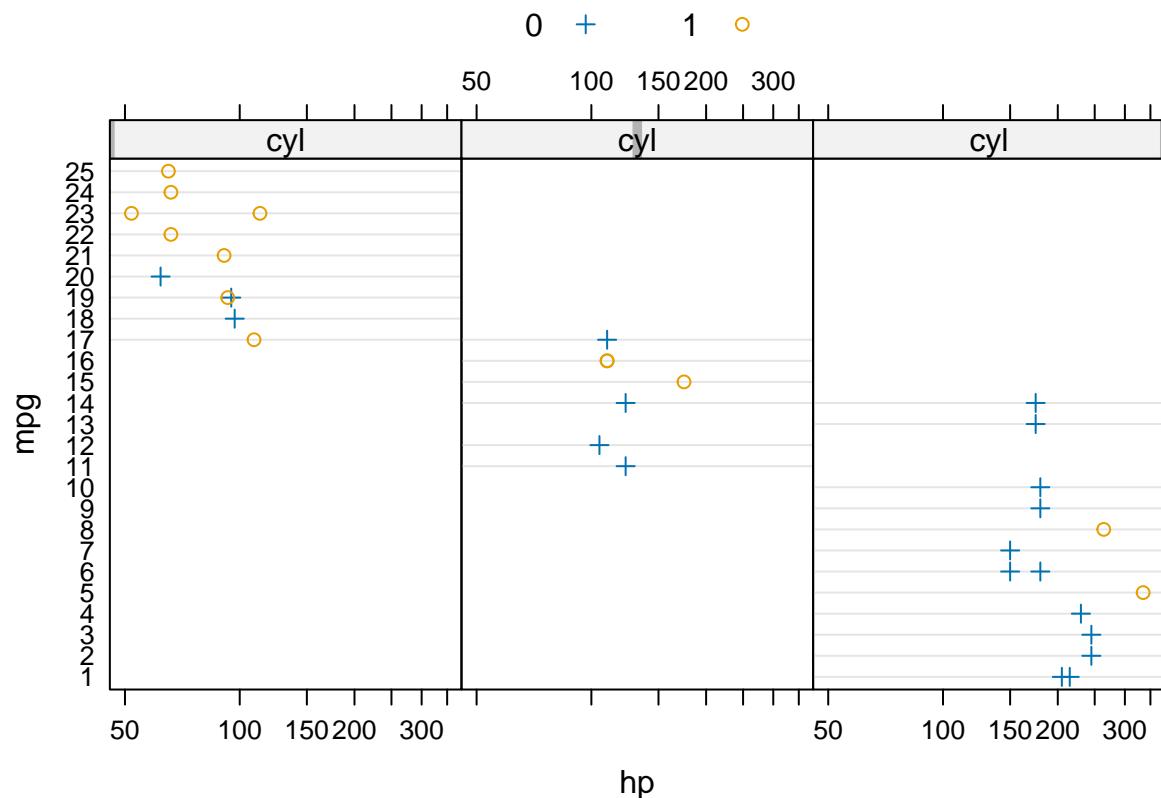
```
dens_plot[2]
```



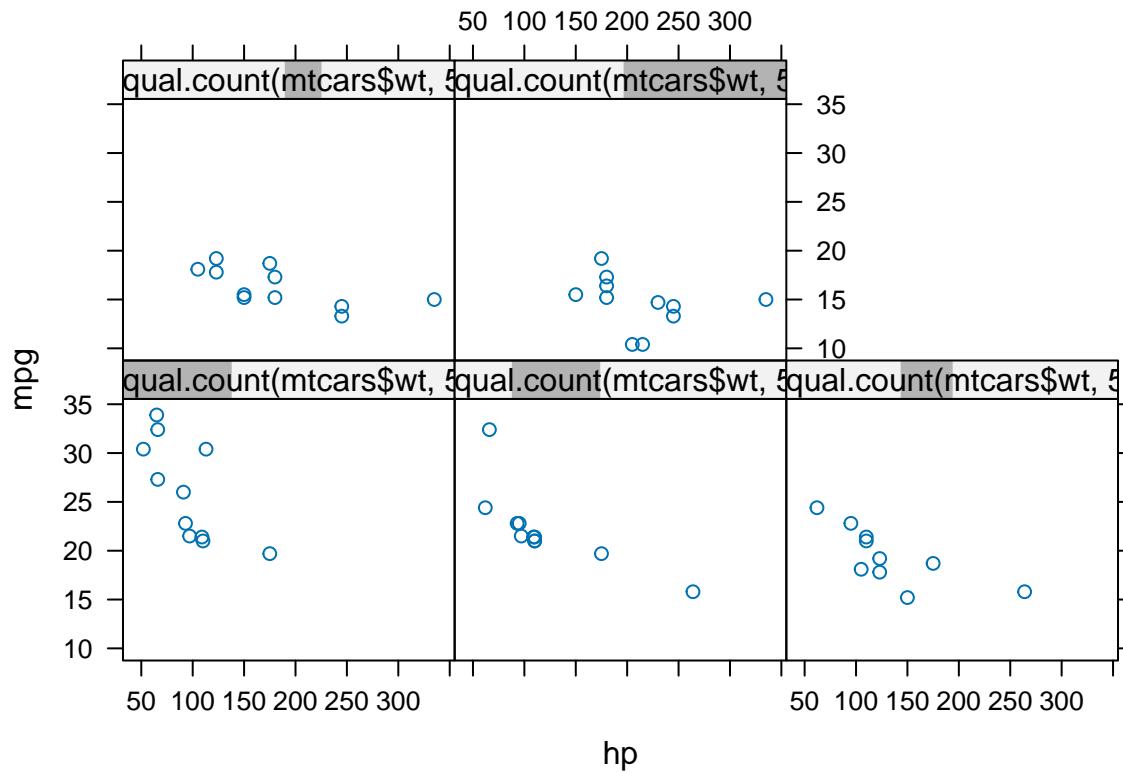
```
#d
dotplot(mpg ~ hp | cyl + am, data = mtcars,
  as.table = TRUE,
  scales = list(x = list(log=TRUE)),
  par.settings = ggplot2like(),
  lattice.options = ggplot2like.opts())
```



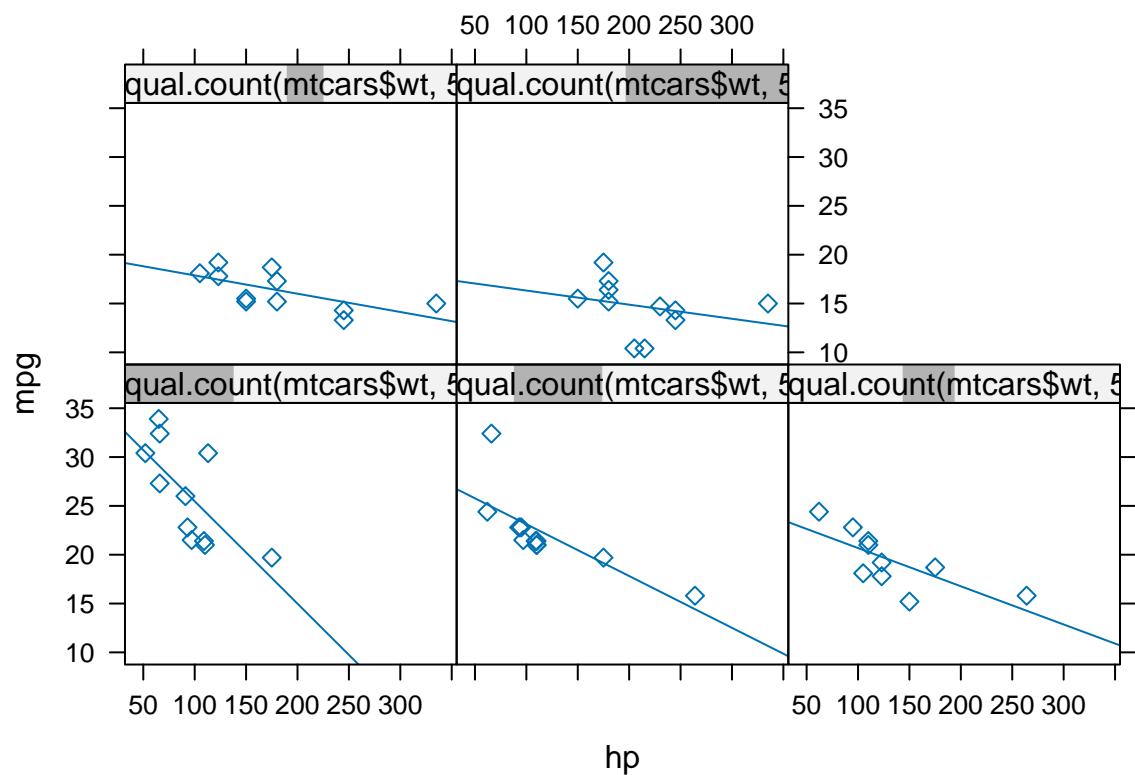
```
#e
dotplot(mpg ~ hp | cyl, data = mtcars,
  groups = am, auto.key = list(columns = 2),
  par.settings = simpleTheme(pch = c(3, 1)),
  scales = list(x = list(log = 2, equispaced.log = FALSE)))
```



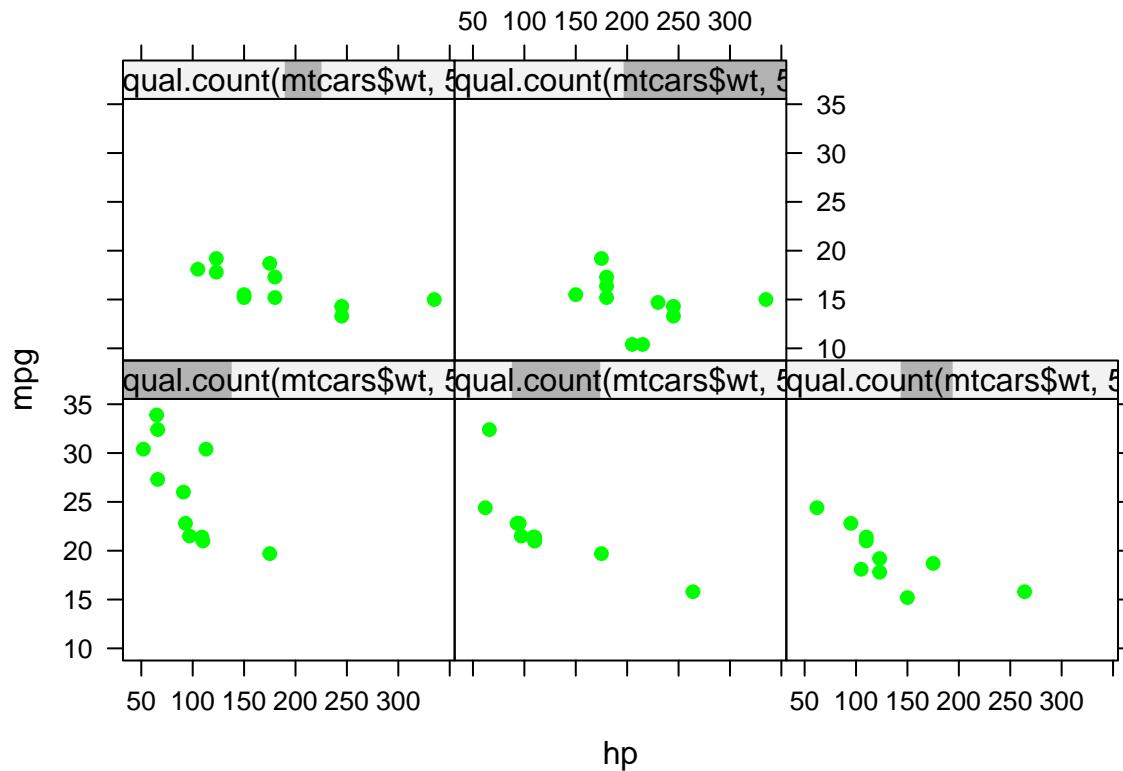
```
#f
plot2<-xyplot(mpg~hp|equal.count(mtcars$wt, 5), data=mtcars)
print(plot2)
```



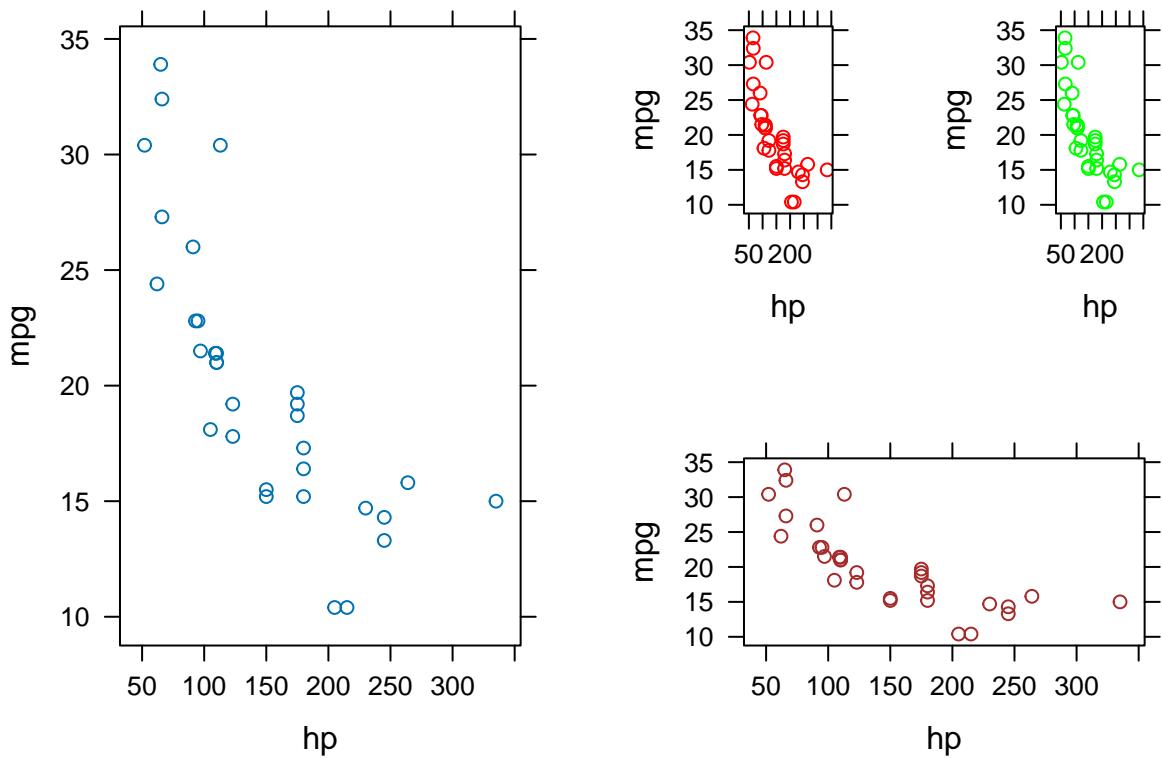
```
plot3<-update(plot2, type=c("p","r"), pch=5)
plot(plot3)
```



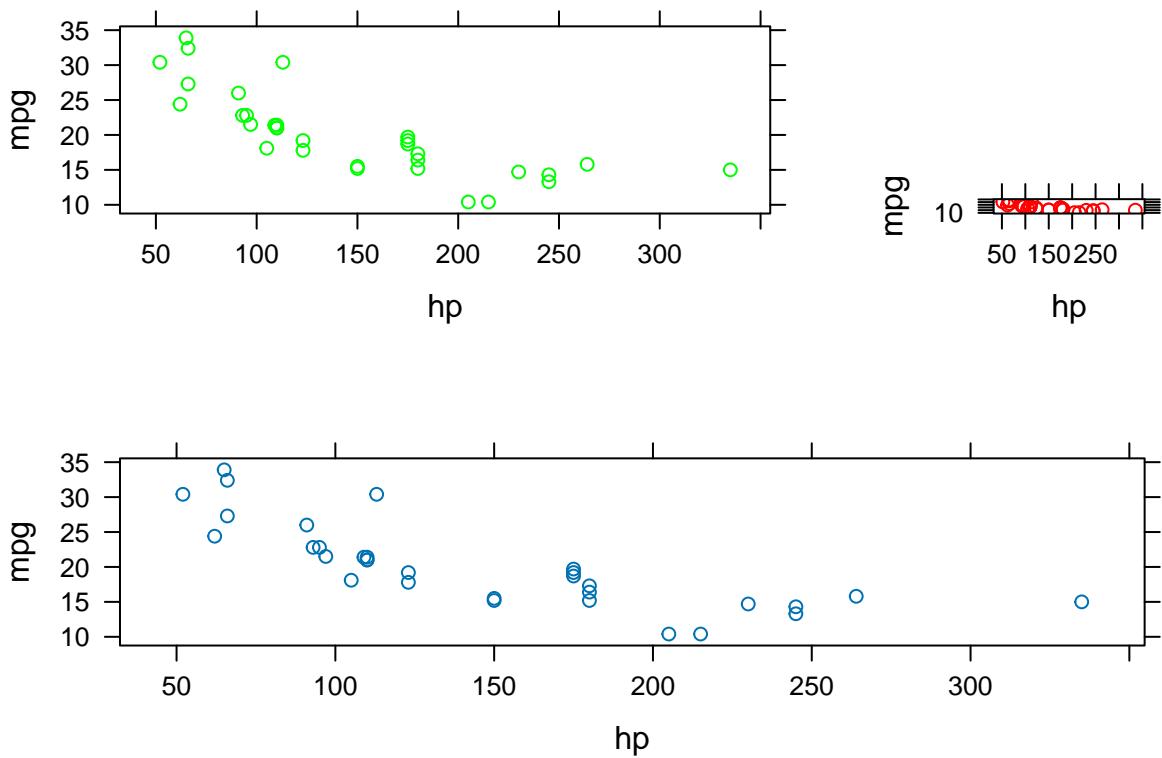
```
update(plot2, pch=19, col="green")
```



```
#g
plot3<-xyplot(mpg~hp, data=mtcars)
plot(plot3, split=c(1,1,2,1))
plot(update(plot3, col="red"), split=c(3,1,4,2), newpage=FALSE)
plot(update(plot3, col="green"), split=c(4,1,4,2), newpage=FALSE)
plot(update(plot3, col="brown"), split=c(2,2,2,2), newpage=FALSE)
```



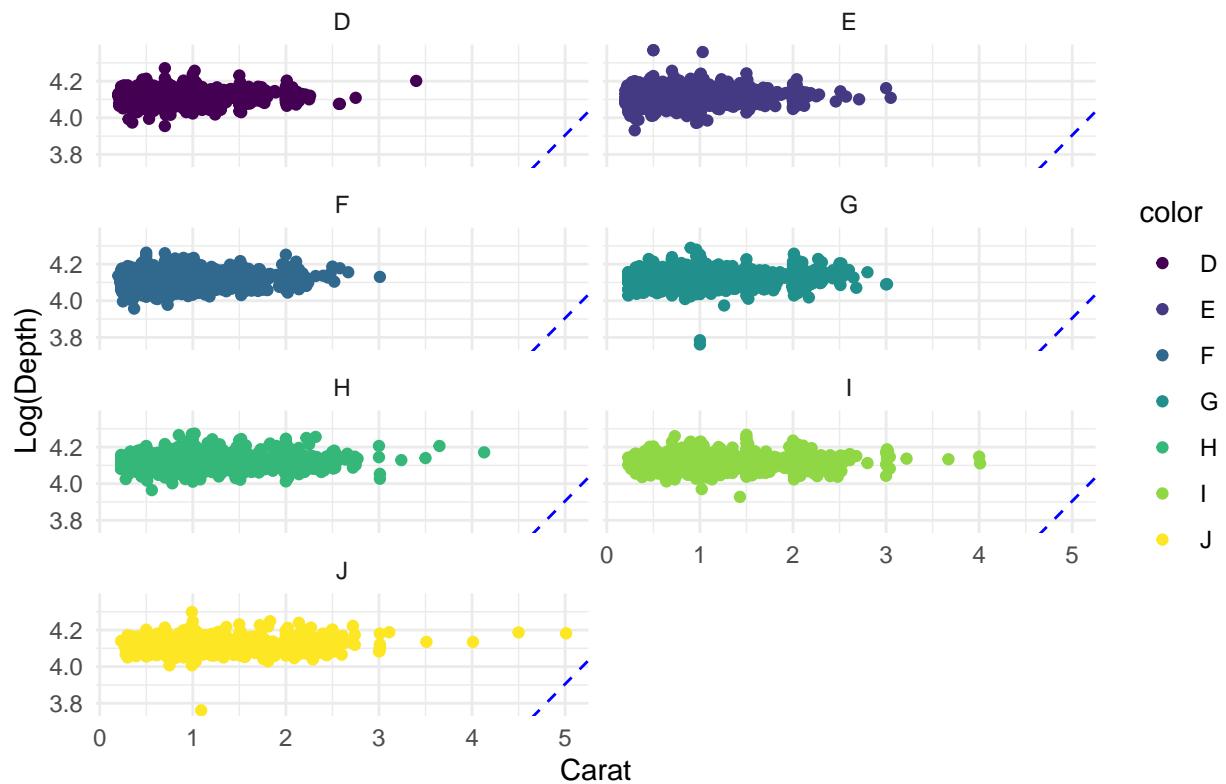
```
#h
plot(plot3, position=c(0,0,1,.5))
plot(update(plot3, col="red"), position=c(0.7,0.5,1,0.8), newpage=FALSE)
plot(update(plot3, col="green"), position=c(0.0, 0.5, 0.7,1),newpage=FALSE)
```



Task 2

```
#a
ggplot(diamonds, aes(x = carat, y = log(depth), color = color)) +
  geom_point() +
  geom_abline(intercept = 1.33, slope = 0.515, linetype = "dashed", color = "blue") +
  facet_wrap(~ color, ncol = 2) +
  theme_minimal() +
  labs(title = "Scatterplot of Carat vs Log(Depth)",
       x = "Carat", y = "Log(Depth)")
```

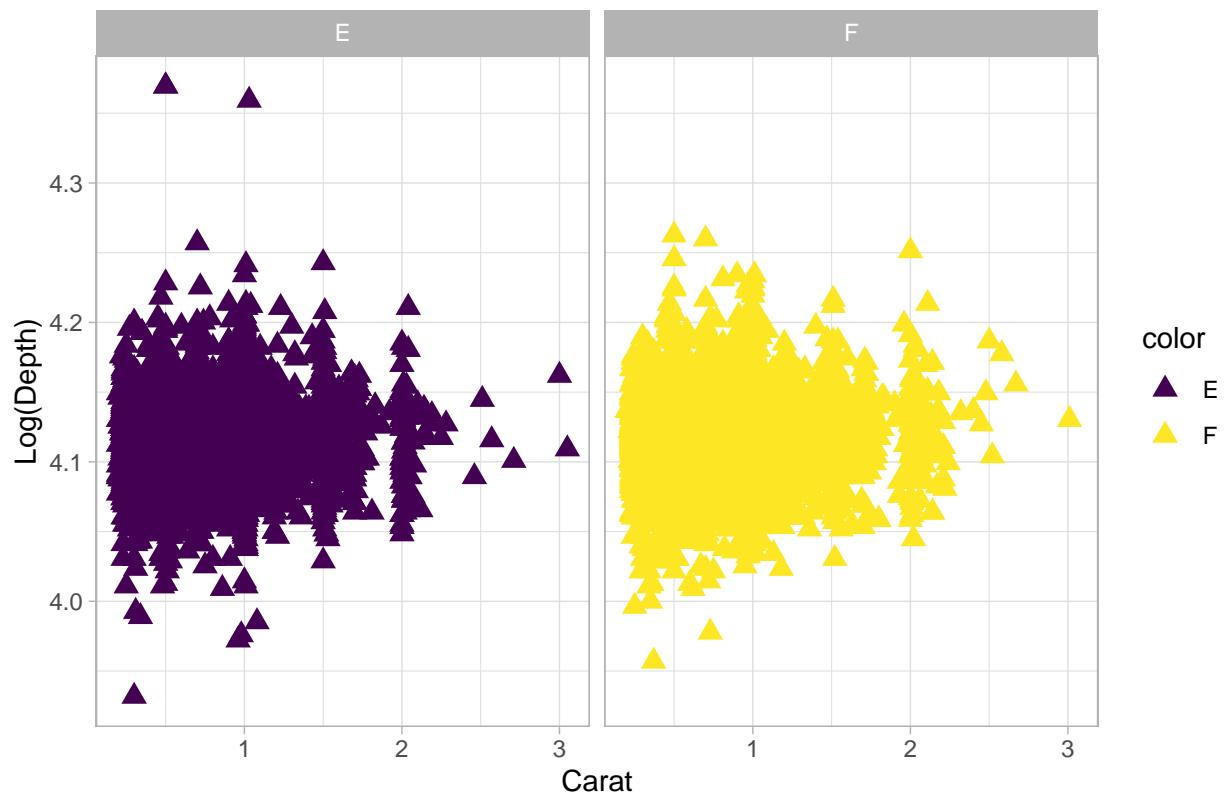
Scatterplot of Carat vs Log(Depth)



```
#b
diamonds_filtered <- diamonds %>% filter(color %in% c("E", "F"))

ggplot(diamonds_filtered, aes(x = carat, y = log(depth), color = color)) +
  geom_point(size = 3, shape = 17) + # Zmieniono wielkość i kształt punktów
  facet_wrap(~ color, ncol = 2) +
  theme_light() +
  labs(title = "Scatterplot of Carat vs Log(Depth) for Colors E and F",
       x = "Carat", y = "Log(Depth)")
```

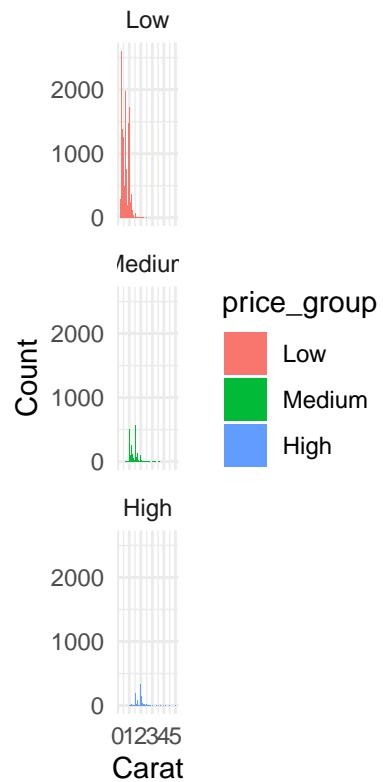
Scatterplot of Carat vs Log(Depth) for Colors E and F



```
#c
diamonds <- diamonds %>%
  mutate(price_group = cut(price, breaks = 3, labels = c("Low", "Medium", "High")))

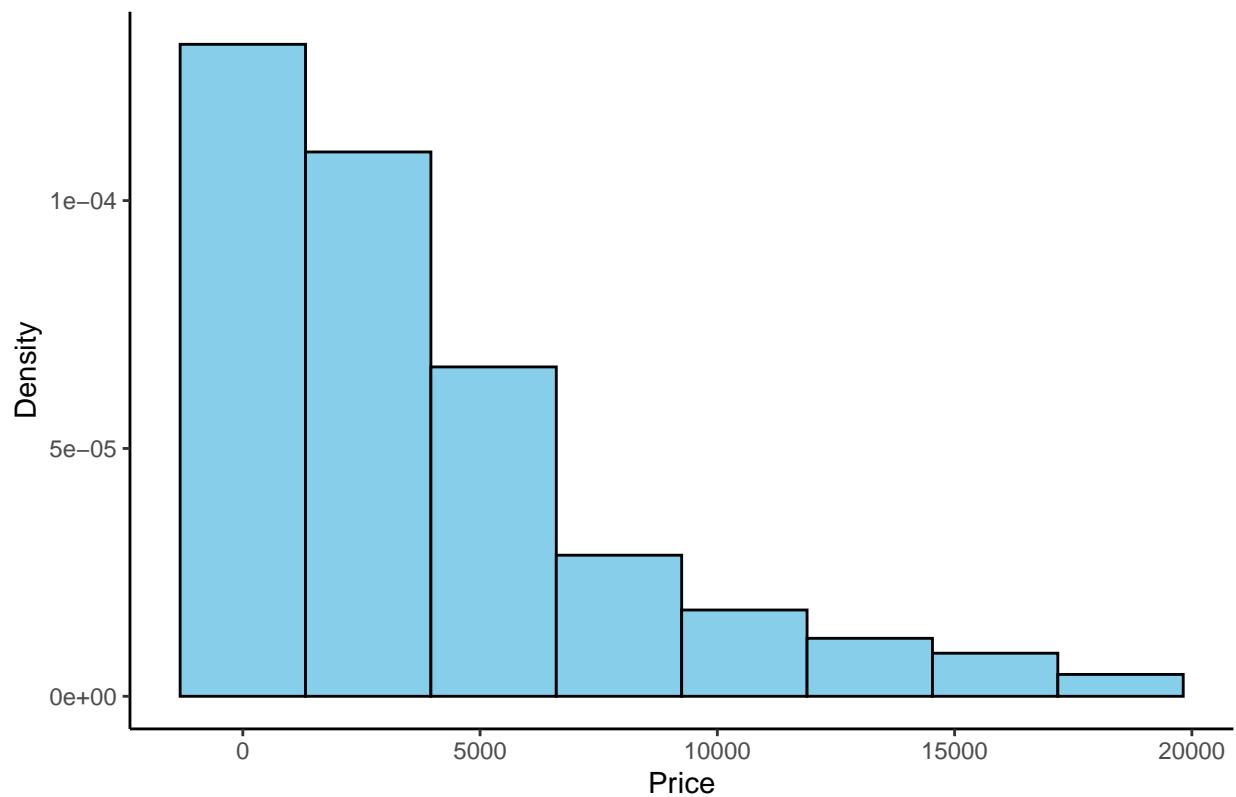
ggplot(diamonds, aes(x = carat, fill = price_group)) +
  geom_bar(position = "dodge") +
  facet_wrap(~ price_group, ncol = 1) +
  theme_minimal() +
  labs(title = "Bar Plot of Carat by Price Group",
       x = "Carat", y = "Count") +
  theme(aspect.ratio = 3)
```

Bar Plot of Carat by Price Group



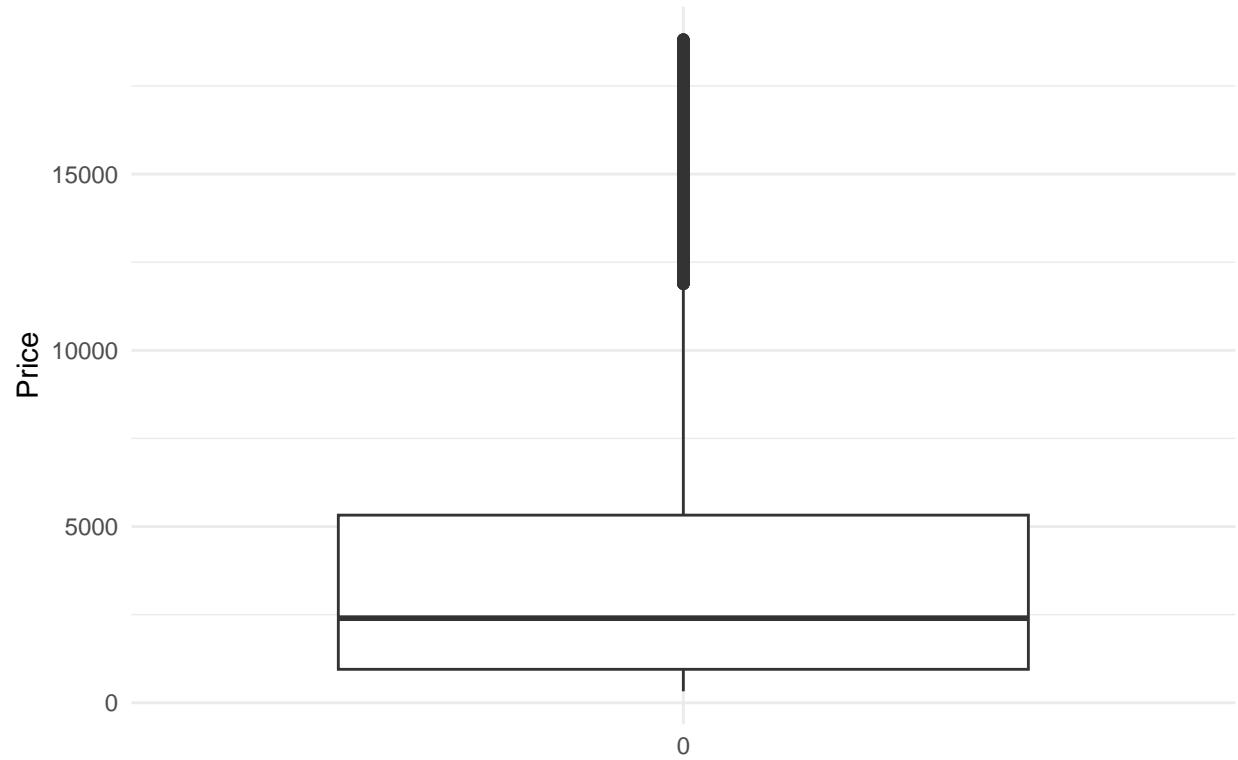
```
#d
ggplot(diamonds, aes(x = price)) +
  geom_histogram(bins = 8, aes(y = ..density..), fill = "skyblue", color = "black") +
  theme_classic() +
  labs(title = "Histogram of Price", x = "Price", y = "Density")
```

Histogram of Price



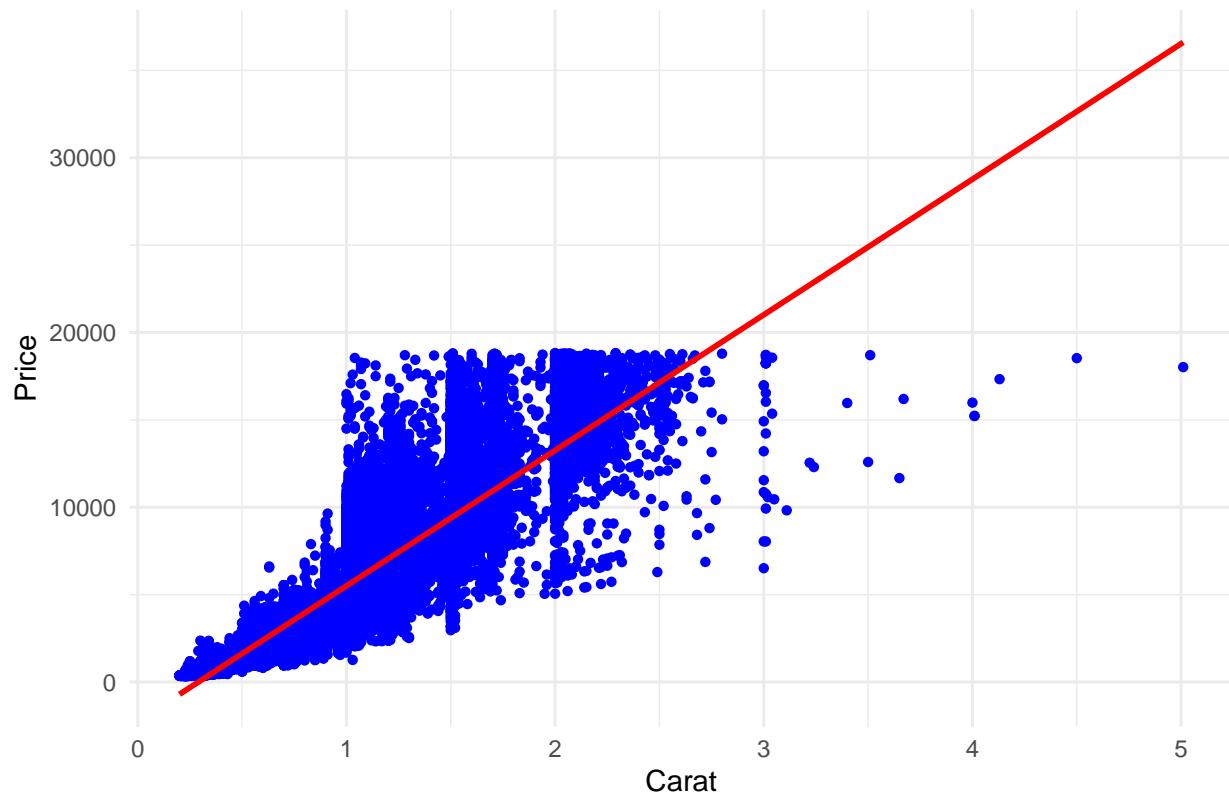
```
#e
ggplot(diamonds, aes(x = factor(0), y = price)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Boxplot of Price", x = "", y = "Price")
```

Boxplot of Price



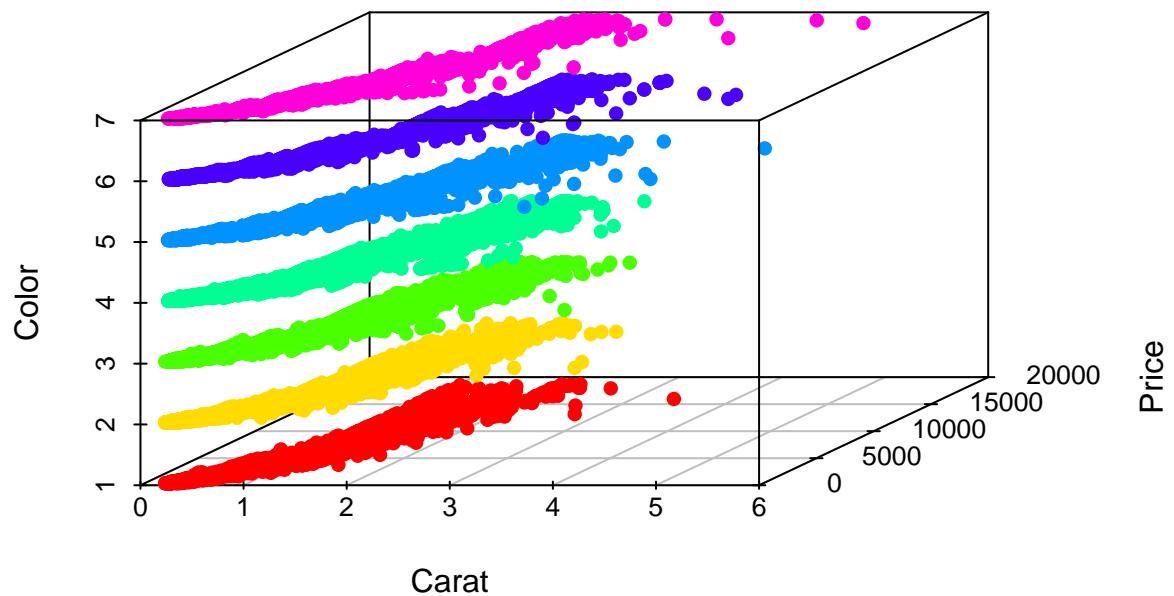
```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point(shape = 16, color = "blue") +  
  geom_smooth(method = "lm", se = FALSE, color = "red") +  
  theme_minimal() +  
  labs(title = "Scatterplot of Price vs Carat", x = "Carat", y = "Price")  
  
## 'geom_smooth()' using formula = 'y ~ x'
```

Scatterplot of Price vs Carat



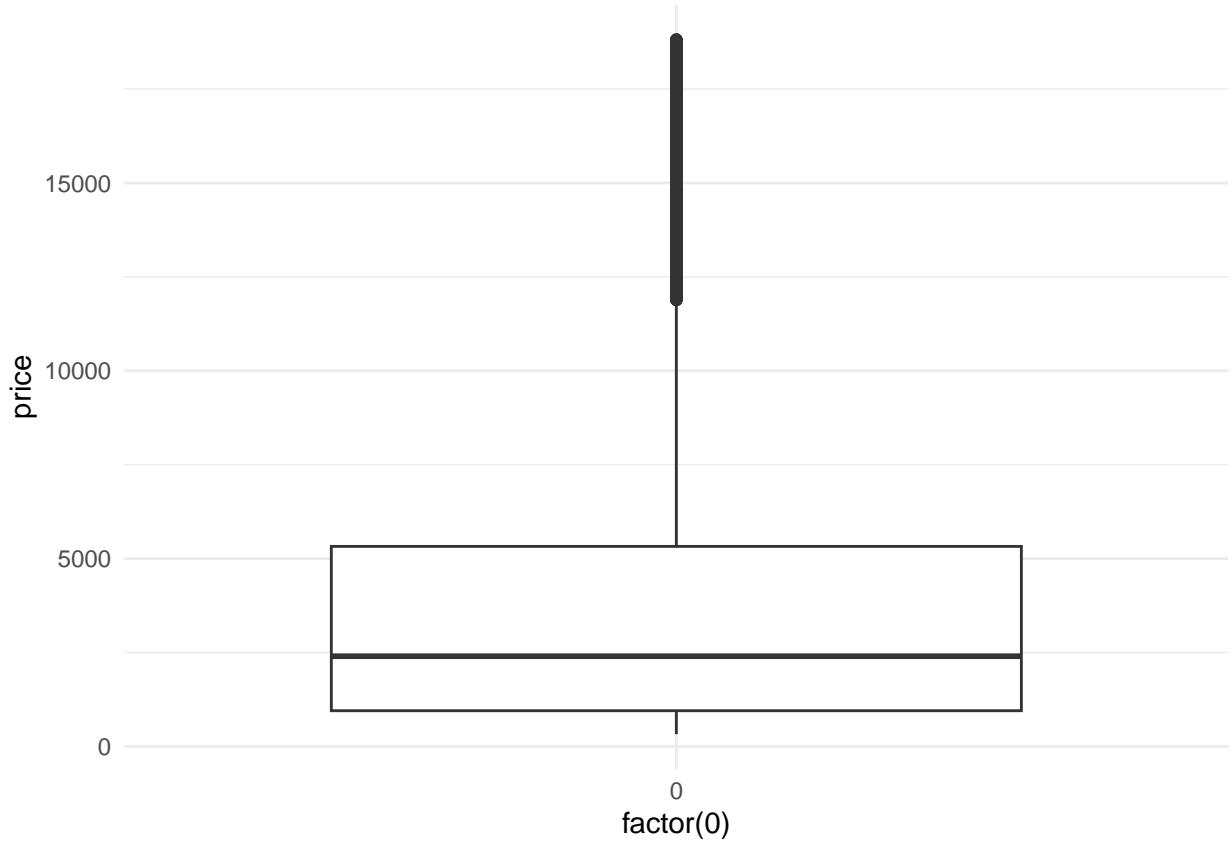
```
library(scatterplot3d)

scatterplot3d(diamonds$carat, diamonds$price, as.numeric(diamonds$color),
              pch = 16, color = rainbow(7)[as.numeric(diamonds$color)],
              xlab = "Carat", ylab = "Price", zlab = "Color")
```



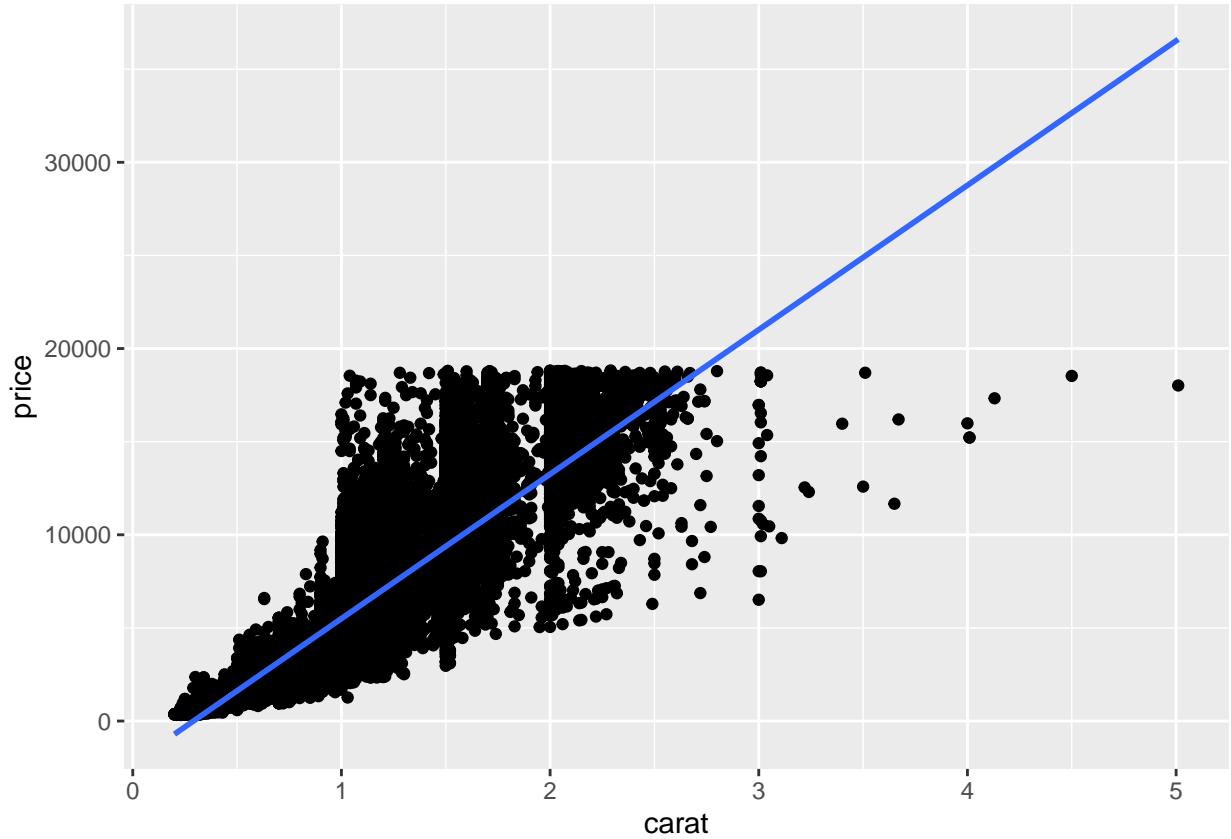
```
#f
layout(matrix(c(1, 2, 2, 3), ncol = 2, byrow = TRUE))

ggplot(diamonds, aes(x = factor(0), y = price)) +
  geom_boxplot() + theme_minimal()
```



```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point() + geom_smooth(method = "lm", se = FALSE)
```

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



```
scatterplot3d(diamonds$carat, diamonds$price, as.numeric(diamonds$color))
```

