



A Linguagem de Programação R: Aplicação em estudos florestais

MSc. José Wesley Lima Silva

Gráficos com ggplot2

AULA 02

Recife, 29 de Novembro de 2019

- 1 Gramática dos gráficos
 - O que é um gráfico?
 - As camadas de um gráfico

- 2 Tipos de gráficos
 - Scatterplot
 - Histograma
 - Barras
 - Boxplots

- 3 Personalizando
 - Escalas e Labels
 - Faces
 - Temas

- 4 Extensões
 - Matriz de correlação
 - Zoom
 - Equações

O que é um gráfico?

O que é um gráfico?

- Um gráfico é o mapeamento dos dados a partir de atributos estéticos (posição, cor, forma, tamanho) de objetos geométricos (pontos, linhas, barras, caixas) (Leland Wilkinson, 2005);

O que é um gráfico?

- Um gráfico é o mapeamento dos dados a partir de atributos estéticos (posição, cor, forma, tamanho) de objetos geométricos (pontos, linhas, barras, caixas) (Leland Wilkinson, 2005);
- Como organizar a criação de gráficos?

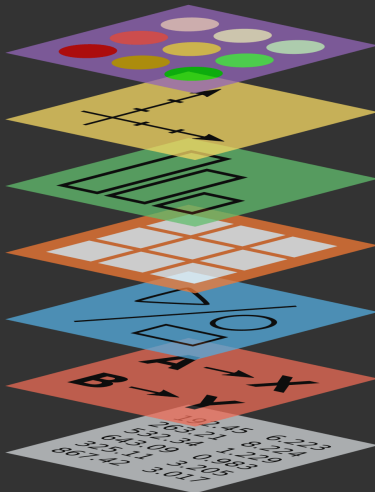
O que é um gráfico?

- Um gráfico é o mapeamento dos dados a partir de atributos estéticos (posição, cor, forma, tamanho) de objetos geométricos (pontos, linhas, barras, caixas) (**Leland Wilkinson, 2005**);
- Como organizar a criação de gráficos?
- **Gramática dos gráficos**: No contexto da análise gráfica, são regras para a construção de gráficos matemáticos e sua representação estética;

O que é um gráfico?

- Um gráfico é o mapeamento dos dados a partir de atributos estéticos (posição, cor, forma, tamanho) de objetos geométricos (pontos, linhas, barras, caixas) (Leland Wilkinson, 2005);
- Como organizar a criação de gráficos?
- **Gramática dos gráficos**: No contexto da análise gráfica, são regras para a construção de gráficos matemáticos e sua representação estética;
- O **ggplot2** é um pacote do R voltado para a criação de gráficos estatísticos, utiliza o conceito de gramática dos gráficos.

Theme
Coordinates
Statistics
Facets
Geometries
Aesthetics
Data



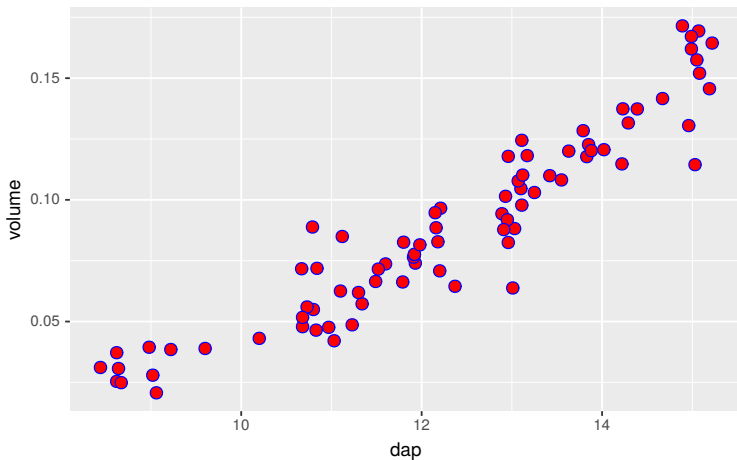
As camadas de um gráfico

- **Data:** Os dados a serem utilizados no formato de **data frame**;
- **Aesthetics:** mapeamento de dados para propriedades estéticas (cor, forma, tamanho, etc.);
- **Geometries:** forma de visualização (pontos, linhas, barras, etc.);
- **Facets:** Uso de Múltiplos gráficos categorizados;
- **Statistics:** Transformações estatísticas usadas (identidade, média, variância, suas variações, regressão linear, etc); **Coordinates:** Mapeamento dos dados no sistema de coordenadas de um plano;
- **Theme:** Temas prapadronizados de cores, tamanhos, letras, etc.

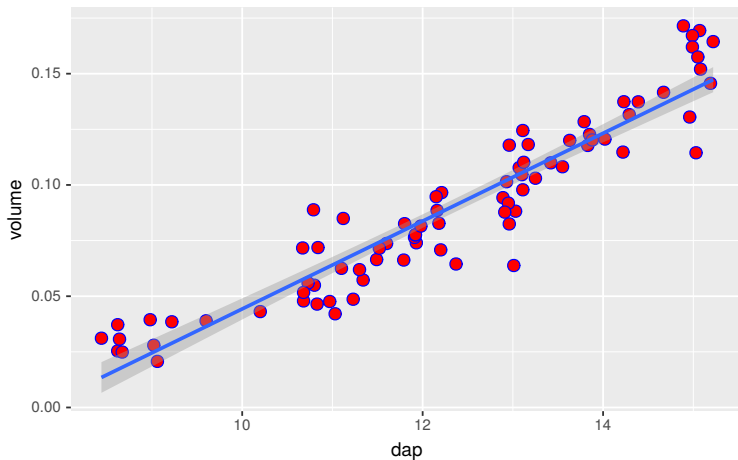
Scatterplot - Dispersão

```
1 #packages
2 library(ggplot2)
3 library(gridExtra)
4
5 #dataset
6 df <- read.csv("base_dados_trat.csv", h=T)
7
8 #scatterplot simples
9 graph.1 <- ggplot(data=df, aes(x=dap, y=volume))
10
11 graph.1 <- graph.1 +
12   geom_point(colour = "blue", size = 3, shape = 21, fill = "red")
13
14 #scatterplot com destaque para os espaçamentos
15 graph.2 <- ggplot(data=df, aes(x=dap, y=volume, color=espaçamento)) +
16   geom_point()
17 #diferença de categorias pode ser utilizada em color, shape, size
18
19 #scatterplot com regressão linear
20 graph.1 + geom_smooth(method = "lm")
21
22 #salvando um gráfico
23 ggsave("graph_ggsave.pdf", width = 10, height = 10, units = "cm", dpi = 900)
```

Scatterplot - Dispersão



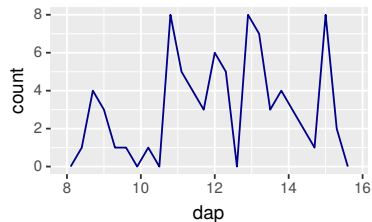
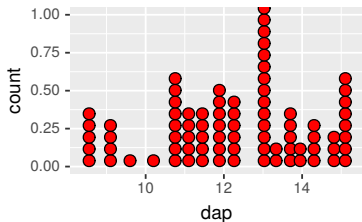
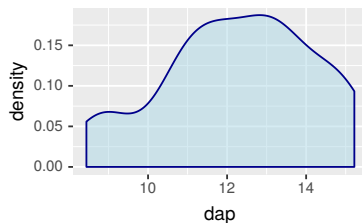
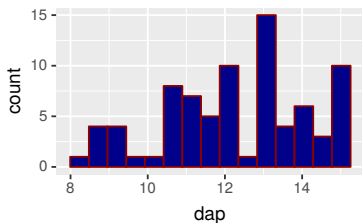
Scatterplot + Regressão Linear



Histograma, Densidade, Dotplot e Polígono de Frequência

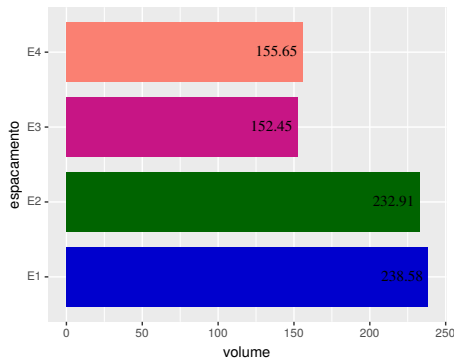
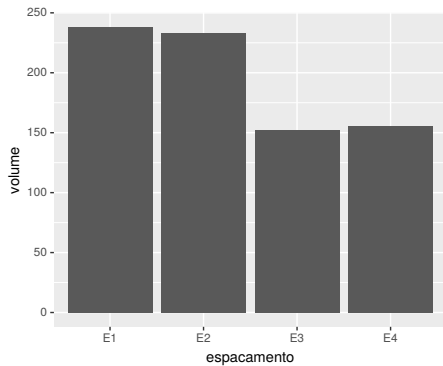
```
1 ##histograma
2 #frequencia
3 graph.3 <- ggplot(data=df, aes(x=dap)) # y = ..count..
4 graph.4 <- graph.3 +
5   geom_histogram(bins=15, colour="darkred", fill="darkblue")
6
7 #densidade
8 graph.5 <- graph.3 +
9   geom_density(alpha=0.5,color="darkblue", fill="lightblue")
10
11 #dotplot
12 graph.6 <- graph.3 +
13   geom_dotplot(binwidth = .3, fill="red")
14
15 #poligono de frecuencia
16 graph.7 <- graph.3 +
17   geom_freqpoly(binwidth = .3, colour="darkblue")
18
19 #grid extra - multiplos graficos
20 grid.arrange(graph.4, graph.5, graph.6, graph.7)
```

Histograma, Densidade, Dotplot e Polígono de Frequência



Barras, Colunas

```
1 #preparando os dados
2 vol.e <- tapply(df$volume, df$espacamento, mean)
3 arv.ha <- c(5000, 2500, 1667, 1250)
4 df.vol <- data.frame(espacamento=names(vol.e), volume=vol.e*arv.ha)
5
6 #barras
7 graph.b1 <- ggplot(data = df.vol, aes(x=espacamento, y=volume)) +
8   geom_bar(stat = "identity")
9
10 graph.b2 <- ggplot(data = df.vol, aes(x=espacamento, y=volume)) +
11   geom_bar(stat = "identity", fill=c("#0000CD", "#006400", "#C71585", "#FA8072"),
12     , width = .8) +
13   geom_text(aes(label=round(volume, 2)), vjust=.5, hjust=1.1, colour='black',
14     family="Times", size=4) +
15   coord_flip()
16
17 #Salvando grafico a partir de um grid.arrange
18 g1 <- grid.arrange(graph.b1, graph.b2, ncol=2)
19 g1 <- arrangeGrob(g1)
20 ggsave("graph_bar.pdf", width = 10, height = 10, units = "cm", dpi = 900, g1)
```



Por categorias

```
1 #densdade por categoria
```

```
2 graph.c1 <- ggplot(data=df, aes(x=dap, fill=espacamento)) +  
3 geom_density(alpha=.5)
```

```
4  
5  
6 #box-plots
```

```
7  
8 graph.c <- ggplot(data=df, aes(y=volume, x=espacamento, fill=espacamento))
```

```
9  
10 graph.c2 <- graph.c +  
11 geom_dotplot(binwidth = .006, binaxis = "y", stackdir = "center")
```

```
12  
13 #boxplot
```

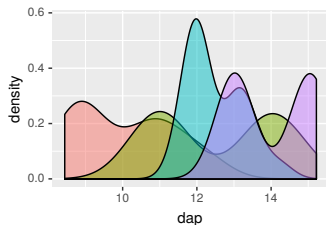
```
14 graph.c3 <- graph.c +  
15 geom_boxplot()
```

```
16  
17 #violin
```

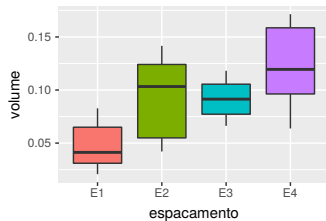
```
18 graph.c4 <- graph.c +  
19 geom_violin()
```

```
20  
21 grid.arrange(graph.c1, graph.c2, graph.c3, graph.c4)
```

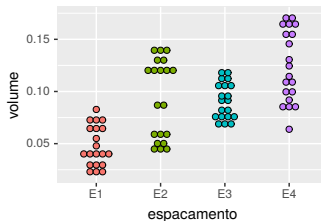
Categorias



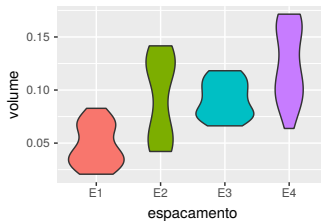
espacamento



espacamento



espacamento



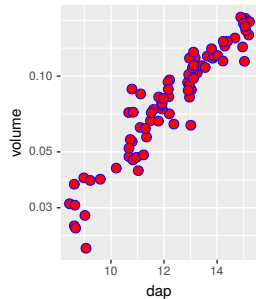
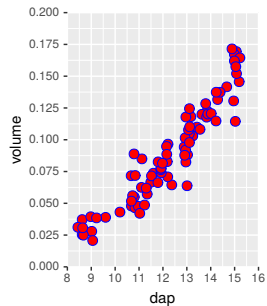
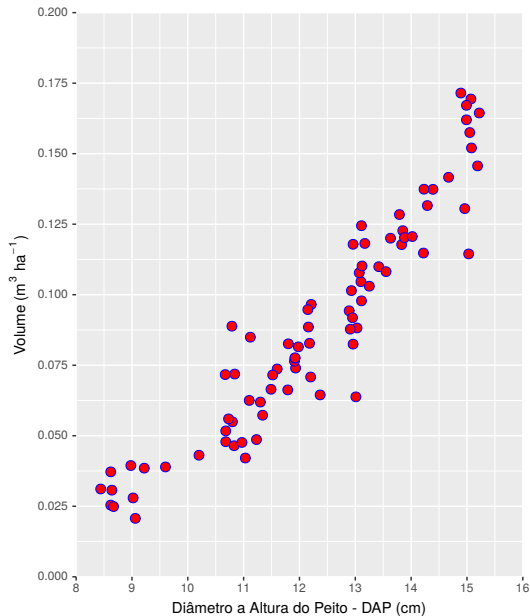
espacamento



Escalas e Labels

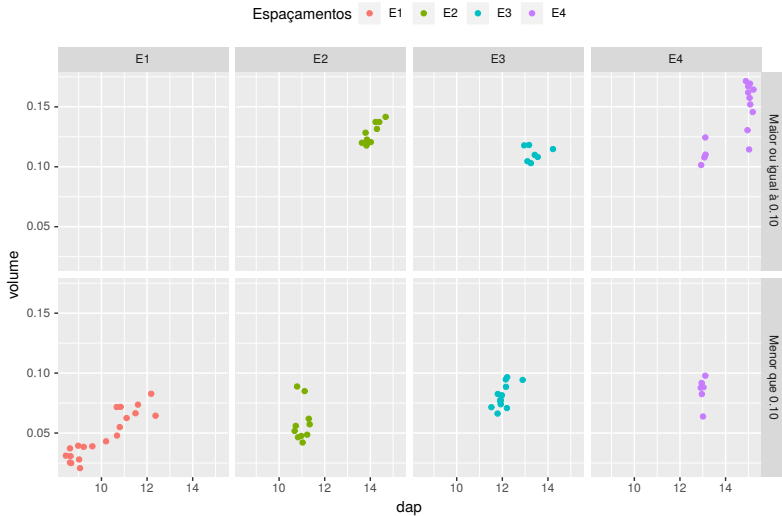
```
1 #Personalizando
2
3 ##escalas
4
5 #limites, breaks, expand
6 graph.p1 <- graph.1 +
7   scale_x_continuous(limits = c(8, 16), breaks = seq(8, 16, 1), expand = c(0,
8     0)) +
9   scale_y_continuous(limits = c(0,0.2), breaks = seq(0, 0.2, 0.025), expand =
10     c(0,0))
11
12 #aplicando transformacoes nos dados
13 graph.p2 <- graph.1 +
14   scale_y_log10()
15
16 ##Labels e expressions
17 graph.p3 <- graph.p1 +
18   xlab("Diámetro a Altura do Peito - DAP (cm)") +
19   ylab(expression("Volume (m"^(3)*" ha"^(1)*""))
20
21 grid.arrange(arrangeGrob(graph.p3), arrangeGrob(graph.p1,graph.p2, ncol=1),
22   ncol=2, widths=c(2,1))
```

Escalas e Labels



Faces

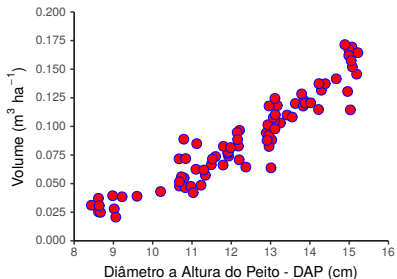
```
1  ###Faces
2
3  graph.f <- ggplot(data=df, aes(x=dap, y=volume, group=espacamento)) +
4    geom_point(aes(color=espacamento))
5
6  graph.f + facet_grid(espacamento ~.) #Em linhas
7  graph.f + facet_grid(.~espacamento) #Em colunas
8  graph.f + facet_wrap(~espacamento) #Em retangulo
9
10 df$vol_cond <- ifelse(df$volume>=.10,"Maior ou igual a 0.10","Menor que 0.10")
11
12 graph.fl <- ggplot(data=df, aes(x=dap, y=volume, group=vol_cond)) +
13   geom_point(aes(colour=espacamento))
14
15 graph.fl + facet_grid(vol_cond~espacamento) + #linhas e colunas
16   scale_colour_discrete(name="Espacamentos") +
17   theme(legend.position = "top")
```



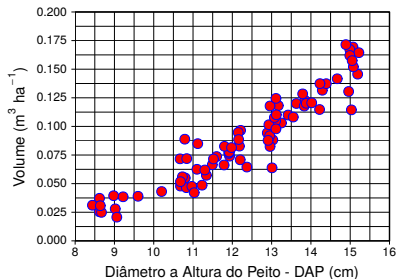
Theme

```
1  ###Temas
2
3  graph.t1 <- graph.p3 +
4    ggtitle("Tema classic") +
5    theme_classic() +
6    theme(plot.title = element_text(hjust = 0.5))
7
8  graph.t2 <- graph.p3 +
9    ggtitle("Tema linedraw") +
10   theme_linedraw() +
11   theme(plot.title = element_text(hjust = 0.5))
12
13 graph.t3 <- graph.p3 +
14   ggtitle("Tema minimal") +
15   theme_minimal() +
16   theme(plot.title = element_text(hjust = 0.5))
17
18 graph.t4 <- graph.p3 +
19   ggtitle("Tema dark") +
20   theme_dark() +
21   theme(plot.title = element_text(hjust = 0.5))
22
23 grid.arrange(graph.t1, graph.t2, graph.t3, graph.t4)
```

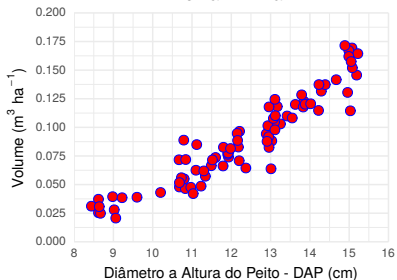
Tema classic



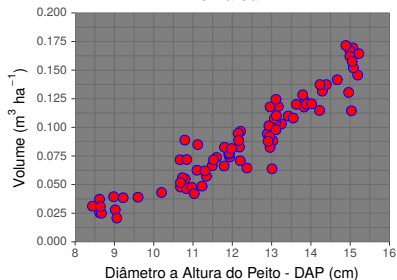
Tema linedraw



Tema minimal



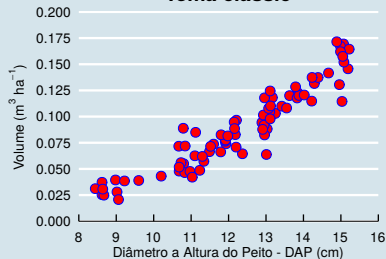
Tema dark



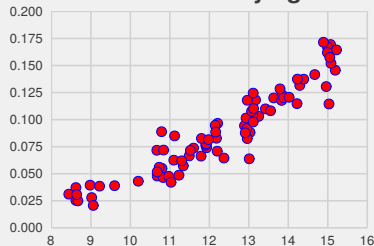
Theme

```
1  ###Temas extras
2  library(ggthemes)
3
4  graph.t5 <- graph.p3 +
5    ggtitle("Tema economist") +
6    theme_economist() +
7    theme(plot.title = element_text(hjust = 0.5))
8
9  graph.t6 <- graph.p3 +
10    ggtitle("Tema fivethirtyeight") +
11    theme_fivethirtyeight() +
12    theme(plot.title = element_text(hjust = 0.5))
13
14  graph.t7 <- graph.p3 +
15    ggtitle("Tema stata") +
16    theme_stata() +
17    theme(plot.title = element_text(hjust = 0.5))
18
19  graph.t8 <- graph.p3 +
20    ggtitle("Tema wsj") +
21    theme_wsj() +
22    theme(plot.title = element_text(hjust = 0.5))
23
24  grid.arrange(graph.t5, graph.t6, graph.t7, graph.t8)
```

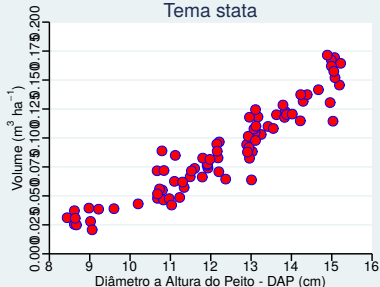
Tema classic



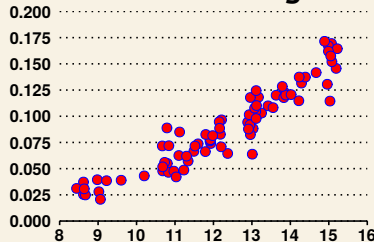
Tema fivethirtyeight



Tema stata



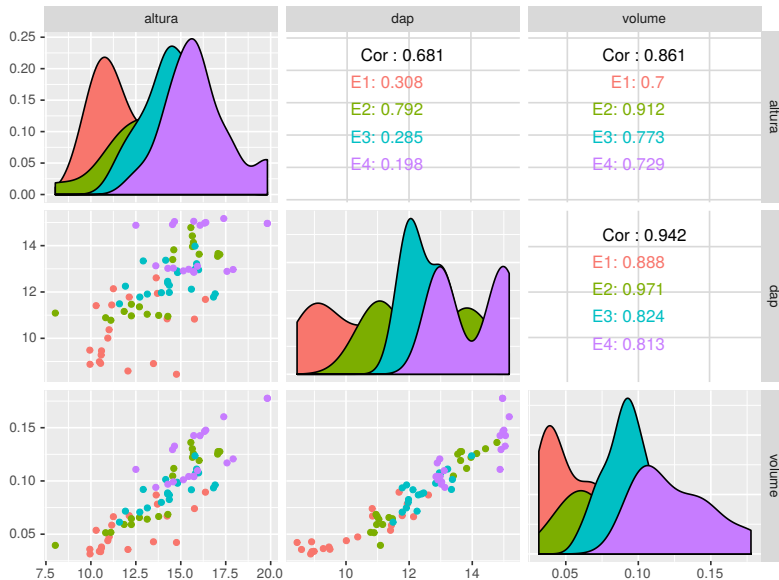
Tema wsj



Matriz de correlação

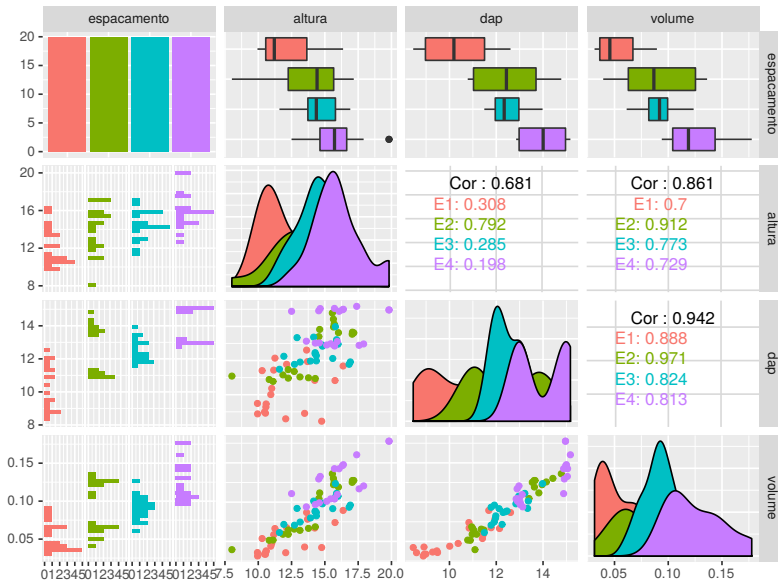
```
1 #GGally
2 #nao instale pelo CRAN - erros no pacote
3 devtools::install_github("ggobi/ggally")
4 library(GGally)
5
6 #dados
7 df.m <- df3 %>%
8   select(espacamento, clone, repeticao, altura, dap, volume)
9
10 ##matriz de correlacao
11
12 graph.m1 <- ggpairs(data = df.m, mapping = aes(color=factor(espacamento)),
13 columns = 4:6, columnLabels = names(df.m)[4:6]) +
14   ggtitle("Matriz de Correlacao") +
15   theme(plot.title = element_text(hjust = .5))
16
17 ##matriz de correlacao com variavel categorica
18
19 graph.m2 <- ggpairs(data = df.m, mapping = aes(color=factor(espacamento)),
20 columns = c(1,4:6), columnLabels = names(df.m)[c(1,4:6)]) +
21   ggtitle("Matriz de Correlacao - Variavel Categorical") +
22   theme(plot.title = element_text(hjust = .5))
```

Matriz de Correlação



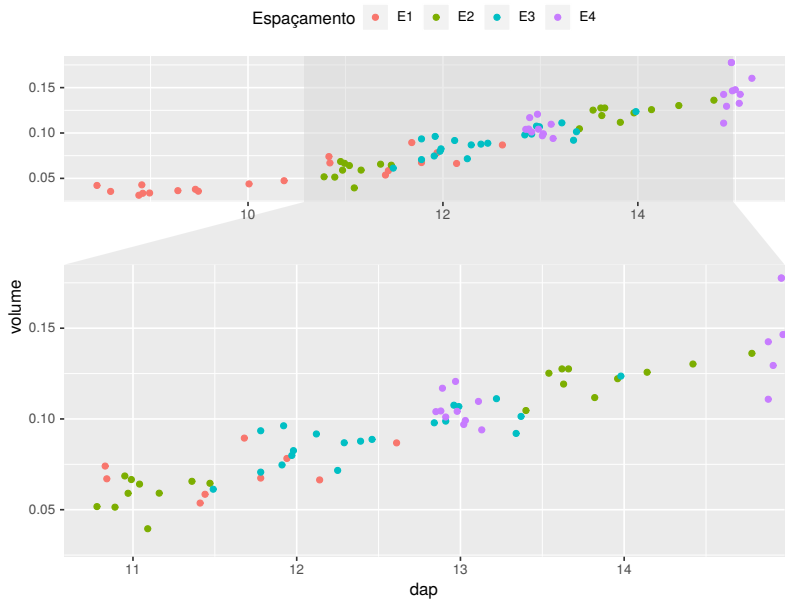
Matriz correlação - Variável categórica

Matriz de Correlação - Variável Categórica



Aplicando Zoom

```
1 library(ggforce)
2
3 graph.z1 <- ggplot(df, aes(x=dap, y=volume, colour=espacamento))+
4   geom_point() + facet_zoom(x = espacamento == "E2") +
5   scale_colour_discrete(name="Espacamento") +
6   theme(legend.position = "top")
```

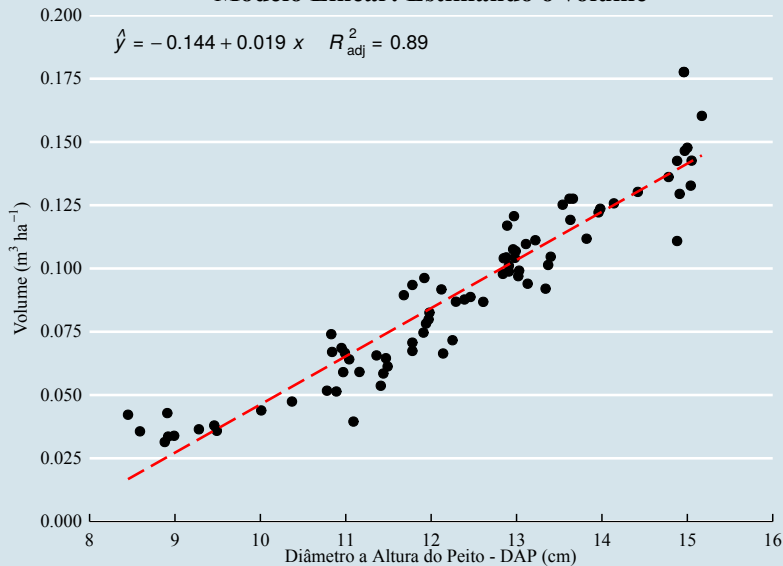


Adicionando uma equação

```

1 #ggpmisc
2 library(ggpmisc)
3
4 modelo.linear <- lm(df$volume~df$dap)
5 summary(modelo.linear)
6 df$vol_pred <- predict(modelo.linear)
7
8 ggplot(data=df, aes(x=dap, y=volume)) +
9   geom_point(shape=21, fill="black", size=2.5) +
10   geom_line(aes(x=dap, y=vol_pred), linetype="longdash", size=.8, color="red") +
11   stat_poly_eq(aes(label = paste(" ", ..eq.label..., ..adj.rr.label..., sep="~~~~"
12     )), formula = modelo.linear, parse=T, eq.with.lhs = "italic(hat(y)) ~ '=' ~",
13     size=4.5) +
14   scale_x_continuous(limits = c(8, 16), breaks=seq(8,16,1), expand=c(0, 0)) +
15   scale_y_continuous(limits=c(0,0.2), breaks=seq(0,.2,.025), expand=c(0,0)) +
16   xlab("âDimetro a Altura do Peito - DAP (cm)") +
17   ylab(expression("Volume (m"^{3}* ha"^{-1}*")")) +
18   ggtitle("Modelo Linear: Estimando o volume") +
19   theme_ws_j(base_family="Times New Roman") +
20   theme(text=element_text(family="Times New Roman", size=16), plot.title =
21     element_text(hjust = .5, size=18))

```


Modelo Linear: Estimando o volume



A Linguagem de Programação R: Aplicação em estudos florestais

MSc. José Wesley Lima Silva

Gráficos com ggplot2

AULA 02

Recife, 29 de Novembro de 2019