**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

*CENTRO DE FILOSOFIA E CIÊNCIAS SOCIAIS*

PÓS-GRADUAÇÃO PSICOLOGIA COGNITIVA

NÚCLEO DE ESTUDOS EM AVALIAÇÃO PISCOLÓGICA

**Doutorandos**

Angélica de Melo Castro

Ricardo Nogueira Maisch

**Professor coordenador**

Dr. José Maurício Haas Bueno

**PARA UM “HELP” NO**

**RECIFE/PE**

**1- INTRODUÇÃO**

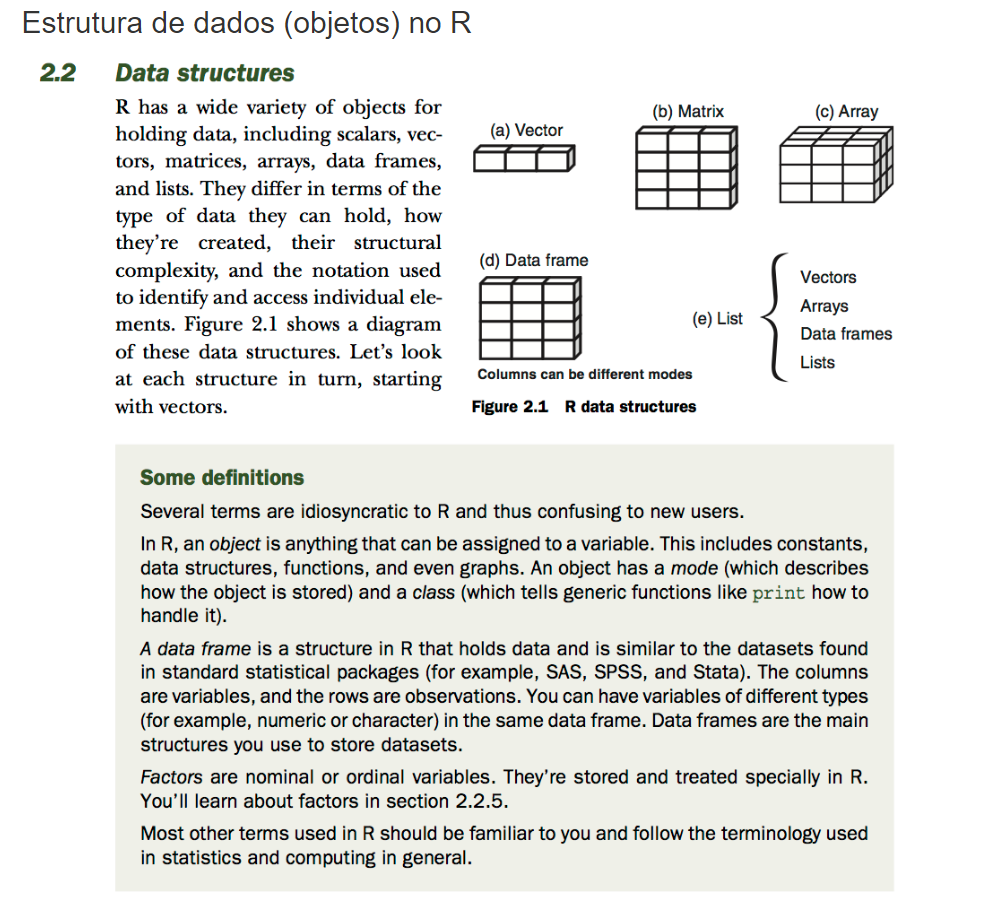
Para iniciar nosso percurso no RStudio, faz-se necessário primeiro responder duas questões que consideramos importantes.

- O que é?

- Como posso instalar?

Para uma visão geral do R, algumas informações são importantes. Primeira é que o R utiliza de linguagem de programação acessível, todas etapas de manipulação dos objetos e funções, em seu sistema, são importantes para sua performance de maneira positiva. O R utiliza-se de linguagem interativa para a manipulação de objetos ou de suas funções. Os objetos podem ser classificados em: conjuntos de dados (data frame), vetores, listas, matrizes ou array.

**Vetores:** conjunto de elementos de uma *mesma natureza*.

**Matrizes:** conjunto de elementos de uma *mesma natureza* organizado em linhas e colunas.

**Array:** generaliza a ideia de matriz. Enquanto em uma matriz os elementos são organizados em duas dimensões (linhas e colunas), em um array os elementos podem ser organizados em um *número arbitrário de dimensões*.

**Data frames:** similar como matrizes, porém diferentes colunas podem possuir elementos de *natureza diferentes***.**

**Listas:** generalizações de vetores, representa uma coleção de objetos.

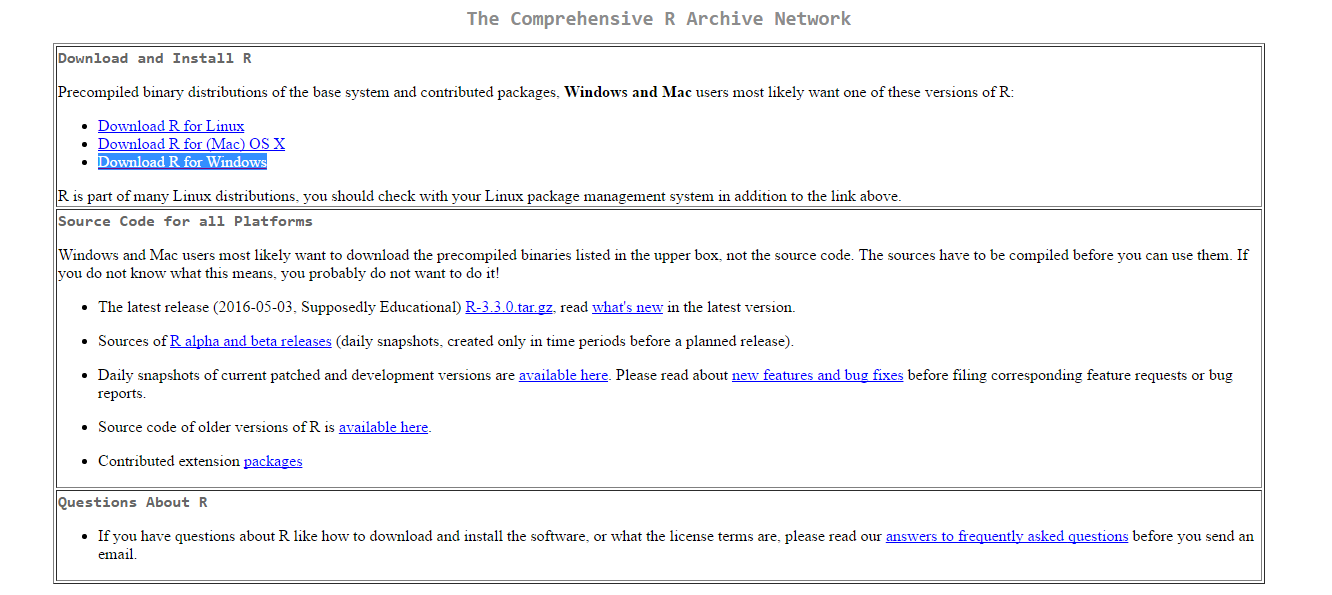
O software é gratuito, possui diversas ferramentas estatísticas disponíveis para análise de dados (objetivo principal para o Neap), cada conjunto de ferramentas para as funções estatísticas estão em pacotes, que precisam ser baixados antes da entrada das análises. O software é aberto, algumas funções de calculadora, por exemplo, podem ser executadas facilmente e sua versão é atualizada sempre que algum pesquisador desenvolve uma nova ferramenta. É possível colocar nas referências (caso seja o caso) o autor do referido pacote - mostraremos a seguir como fazer.

Mas, depois desses esclarecimentos gerais, podemos ir para os comandos de instalação do software.

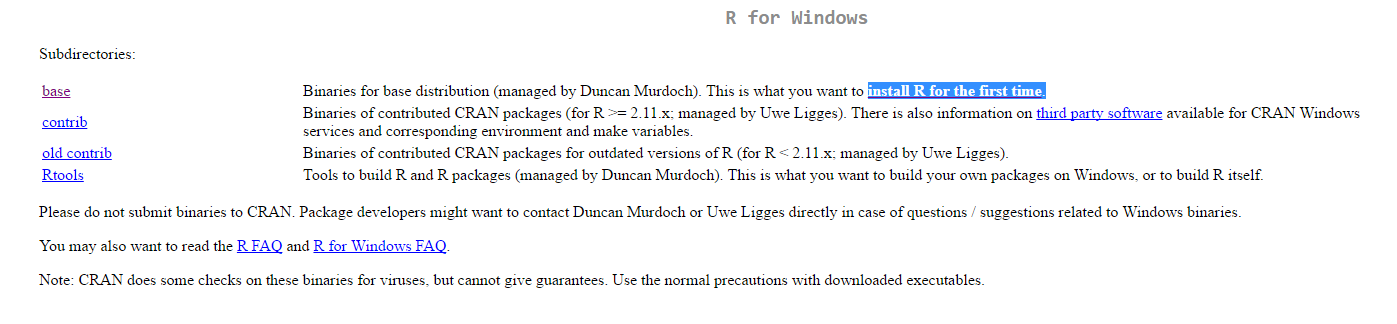
- Como instalar?

O primeiro passo é instalar o R convencional usando o caminho:

<https://cran.r-project.org/mirrors.html>

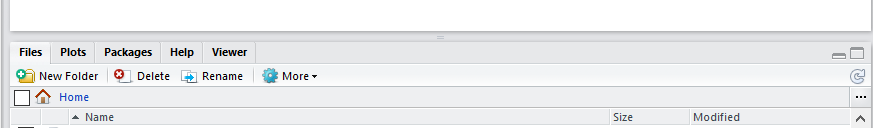
Escolha o País na coluna da esquerda e na coluna da direita a universidade, nós escolhemos a Universidade de São Paulo (USP). 

Quando a tela acima for aberta, clique em seu sistema operacional. Nós escolhemos o link: **Download R for Windows.**

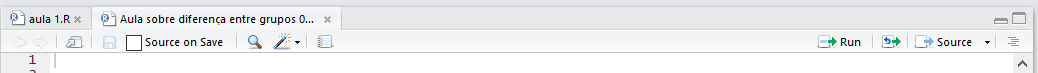
Para a primeira instalação escolha o link da janela a seguir “**install R for the first time**”. Conforme na janela abaixo. Depois clique no link: “**Download R 3.3.0 for Windows**”.

O RStudio é um ambiente de desenvolvimento integrado aberto e livre para o R. Após a instalação do R, acesse o site do Rstudio em : **www.rstudio.org.** Clique no menu *download.*Escolha Download *RStudio Desktop*.Escolha o sistema operacional (Linux, Mac ou Windows). Nós escolhemos o sistema operacional **Windows Vista/7/8/10**. Instale o programa a partir do arquivo salvo.

**2- PRIMEIROS PASSOS**

A[[1]](#footnote-1) tela principal do RStudio é disposta por 4 janelas. No canto superior a esquerda é apresentada a janela *Source*, na qual são disponibilizados os *scripts* (códigos de programação previamente redigidos e salvos em arquivo com extensão .R), arquivos de texto, documentos Sweave, documentação do R e documentos TeX. Na janela superior a direita, a primeira aba é disponibilizada para gerenciar diferentes áreas de trabalho. Já na segunda aba desta janela fica registrado o histórico de todos os *scripts*, funções e ações executadas. Na parte inferior a esquerda, localiza-se a janela do *Console*, a mesma janela padrão disponibilizada no R. E finalmente, na janela inferior a direita, são agrupadas em uma janela outras 4 abas: a primeira delas é um gerenciador de arquivos (aba *File*), na segunda são exibidos os gráficos gerados pelo RStudio (aba *Plots*). Na terceira aba são apresentados os pacotes já instalados (aba *Packages*). E finalmente, a quarta aba trata-se da Ajuda (aba *Help*) do R/RStudio.

Na janela *Source* ao editar o *script* é possível comentar ou não automaticamente (sem necessidade de digitação) determinada parte do código utilizando a opção *Edit/Comment/Uncomment Lines*, mantendo o cursor na linha a ser executada ou selecionar o conjunto de linhas desejado. Nesta tela, também é possível indentar automaticamente comandos, uma prática comum na organização de *scripts* na ciência da computação. Os *scripts* podem ser executados por meio do botão *Run*, não havendo mais necessidade de clicar com o botão direito do *mouse* linha a linha do *script* como é feito no R padrão.



**2.1- O R como calculadora**

O forma de uso mais básica do R é usá-lo como calculadora. Os operadores matemáticos básicos são: + para soma, - subtração, \* multiplicação, / divisão e ^ exponenciação. Digite as seguintes operações na linha de comandos do R:

> 2+2

> 2\*2

> 2/2

> 2-2

> 2^2

Use parênteses para separar partes dos cálculos, por exemplo, para fazer a conta 4+16, dividido por 4, elevado ao quadrado:

> ((4+16)/4)^2

Outros cálculos possíveis:

sqrt() para a raiz quadrada

abs() - valor absoluto

exp() - exponencial

log() - logaritmo na base e

log10() - logaritmo na base 10

OBS.: Você pode utilizar o símbolo (#) para desabilitar os comandos do software e fazer as anotações que desejar. Este mecanismo é importante para que você saiba, num segundo momento, quais passos você percorreu para verificar alguma análise.

**2.2 - Como citar o R, ou um pacote do Rstudio em publicações:**

No R existe um comando que mostra como citar o R ou um de seus pacotes. Veja como fazer:

> citation() > citation(*colocar o nome do pacote instalado*)

Por exemplo:

Digite e rode o comando:

install.packages("readxl")

Agora verifique sua citação, digitando e rodando o seguinte comando:

citation("readxl")

Você encontrará como resultado:

Hadley Wickham (2016). readxl: Read Excel Files. R package version 0.1.1. https://CRAN.R-project.org/package=readxl

**2.3 – Instalando pacotes necessários**

Digite e rode o seguinte comando:

install.packages(c("haven", "readxl", "readr", "xlsx","foreign"))

Ou, você pode atribuir uma variável para todos esses pacotes de uma só vez, utilizando:

v <- c("haven", "readxl", "readr", "xlsx","foreign")

A expressão : v <- c é um comando de concatenação de uma nova variável – a variável v está assumindo todos os pacotes em c (você pode concatenar utilizando a letra que quiser, a letra c é padrão do Rstudio, por isso recomenda-se não utilizar a letra c). Então, uma outra forma de instalar os pacotes dados é:

install.packages(v)

OBS.: Você pode encontrar a citação de cada um dos pacotes instalados.

**2.4 – Como trazer um arquivo para o R**

Para realizar essa tarefa alguns comandos são necessários:

setwd(“") - muda o diretório do trabalho

getwd() - mostra o diretório do trabalho

names() - lista o nome dos objetos criados na sessão atual

str() – mostra a estrutura do arquivo

dir() - lista todos os arquivos na pasta de trabalho atual

search() - lista todos os pacotes carregados

rm()- remove o objeto entre parênteses

rm(list=ls(all=TRUE)) - remove todos os objetos, limpando a memória

attach() - reconhece os objetos dentro de um data frame

detach() – função que desfaz o attach

Para começar a apontar o diretório de trabalho utilize:

setwd("~/Dropbox/Stat R")

O comando dentro das aspas é o nome do arquivo que se está direcionando o programa. Após apontar o diretório, abra o pacote (readxl) instalado, usando:

**library**(readxl)

Com:

df1 <-read\_excel("lego.xlsx", col\_names = TRUE, sheet = "Sheet1")

Você está colocando no **Global Enviroment** um novo **objeto – df1. O pacote aberto lê arquivos do Excel.**

Para importar arquivos do SPSS, abra o pacote:

**library**(haven)

E crie outro objeto no **Global Enviroment** :

df2 <-read\_spss("bd\_geral\_03\_01\_2014.sav")

Agora no **Global Enviroment** possui dois objetos df1 e df2 importados do Excel e outro do SPSS.

Salve os dados do projeto. Note que o R salva todos os arquivos ou objetos listados no **Global Enviroment.** Salvando os objetos em arquivos que podem ser lidos posteriormente:

*#Salva todos objetos*

save.image(file = "t1.RData")

*# Salva somente um objeto*

save(list = "df1", file = "t1.RData")

**2.5 - Explorando os dados nos objetos**

Vendo o nome das variáveis com o comando **names()**

names(df1)

#Faça um, depois o outro e verifique os nomes de todas as variáveis contidas no arquivo.

names(df2)

Veja a estrutura dos arquivos

str(df1)

Alternativamente transforme o arquivo em tabela - tbl class - usando o pacote **dplyr**

*# install.packages("dplyr")*

**library**(dplyr)

##

## Attaching package: 'dplyr'

##

## The following objects are masked from 'package:stats':

##

## filter, lag

##

## The following objects are masked from 'package:base':

##

## intersect, setdiff, setequal, union

df1 <-tbl\_df(df1)

df2 <-tbl\_df(df2)

Veja agora o que acontece quando você simplesmente digita e roda o nome do *dataframe*

df1

Seleção de variáveis. Use o número das variáveis entre [linhas, colunas ] logo após o nome do objeto, nesse caso, um dataframe.

1:5

## [1] 1 2 3 4 5

names(df1[ , 1:5])

## [1] "id" "IDA\_SEXO" "IDA\_DT\_NASCIMENTO"

## [4] "IDA\_SERIE" "IDA\_TURMA"

c(1, 4, 35)

## [1] 1 4 35

names(df1[ , c(1, 4, 35)])

## [1] "id" "IDA\_SERIE" "ise\_2"

*# Elimina variáveis*

df2 <- df2[ , -(3:4)]

*# Criando um novo dataframe com as variáveis selecionadas*

temp <- df1[ , 1:5]

*# removendo da área de trabalho*

rm(temp)

Seleção de observações. É possível usar os números das linhas ou operações lógicas $ e |

*# Seleção usando números*

df1[1:5 , 1:5]

*# Seleção usando lógica*

df1b <- df1[df1$IDA\_SERIE==5, ]

df1b <- df1[df1$IDA\_SERIE>3, ]

df1b <- df1[df1$IDA\_SERIE==5 & df1$IDA\_TURMA=="A", ]

**Exercício 1: importando dados**

* Crie um diretório
* Escolha um banco de dados próprio em excel ou SPSS
* Copie-o no diretório criado
* Abra o RSTUDIO
* Crie um **R Script** para escrever seus programas e salve no diretório criado chamando-o de dados.R
* Aponte o RSTUDIO para o diretório criado
* Importe o arquivo
* Estudo sua estrutura usando o str()
* Salve os dados no diretório

3**- ESTRUTURA DOS DADOS**

Aponte o diretório de trabalho:

setwd("~/Dropbox/Stat R")

Veja com concatenação de variáveis a criação de vetores numéricos, qualitativos ou lógicos.

a <- c(1, 2, 5, 3, 6, -2, 4)

b <- c("one", "two", "three")

c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)

#agora rode separadamente cada um dos vetores

a

b

c

#veja sua classe

class(a)

class(b)

class(c)

#veja sua estrutura usando:

str(a)

str(b)

str(c)

**3.1- Criando uma estrutura tipo Matriz:**

Agora você pode criar matrizes especificando o tipo da matriz com número de linhas e número de colunas:

y <- matrix(1:20, nrow=5, ncol=4)

#Seguindo os mesmos passos utilizados com os vetores, observado sua classificação e se quiser obervar sua estrutura

y

class(y)

#Criando uma matriz qualquer com os seguintes dados:

c <- c(1,26,24,68)

#Atribuindo nomes para linhas, para as colunas e organizando-a começando pelas linhas do comando , byrow=TRUE:

rnames <- c("R1", "R2")

cnames <- c("C1", "C2")

matriz <- matrix(c, nrow=2, ncol=2, byrow=TRUE,

#Aqui a matriz criada:

dimnames=list(rnames, cnames))

**3.2- Criando uma estrutura tipo Arrays:**

dim1 <- c("A1", "A2")

dim2 <- c("B1", "B2", "B3")

dim3 <- c("C1", "C2", "C3", "C4")

#Foi estabelecido acima três dimensões. No comando abaixo você está

Criando a estrutura tipo array.

z <- array(1:24, c(2, 3, 4), dimnames=list(dim1, dim2, dim3))

#Você pode a exemplo dos vetores e das matrizes, verificar sua classe e estrutura.

z

class(z)

**3.3- Criando uma estrutura tipo Data Frame:**

Criando um Dataframe a partir de um exemplo qualquer. Identificação de pacientes, idade, tipo de diabete e estado.

patientID <- c(1, 2, 3, 4)

age <- c(25, 34, 28, 52)

diabetes <- c("Type1", "Type2", "Type1", "Type1")

status <- c("Poor", "Improved", "Excellent", "Poor")

#comando para criar o dataframe

patientdata <- data.frame(patientID, age, diabetes, status)

patientdata

**3.4- Criando fatores:**

status <- c("ruim", "bom", "arrazou", "ruim", "muito bom", "arrazou")

#organizando os fatores como você gostaria:

status <- factor(status, levels=c("ruim", "bom", "muito bom", "arrazou"))

#Agora é possível verificar como ficou a organização utilizando

O comando:

class(status)

levels(status)

#Você também pode organizar utilizando variáveis numéricas:

as.numeric(status)

as.character(status)

#O mesmo pode ser feito criando o fator da variável sexo:

sex <- factor(sex, levels=c(1, 2), labels=c("Male", "Female"))

**3.5 - Criando listas:**

#Nomeando sua lista.

g <- "My First List"

#Estabelecendo as variáveis, no caso idades.

h <- c(25, 26, 18, 39)

#Apontando numa matriz como gostaria que as idades fossem organizadas.

j <- matrix(1:10, nrow=5)

#Apontando quantas matrizes serão encontradas.

k <- c("one", "two", "three")

#Criando a lista em:

mylist <- list(title=g, ages=h, j, k)

#Verificando a lista com os comandos:

class(mylist)

mylist

names(mylist)

#Você pode criar níveis de lista ou apontar para partes específicas de uma lista com os comandos:

#Dando um título

mylist$title

#Especificando em:

mylist[1]

mylist[[1]]

mylist[[3]][ , 2]

## Exercício 2: aprendendo estrutura de dados

* Crie um vetor
* Crie uma matriz
* Crie um dataframe
* Crie uma lista

# **4 – ANÁLISES ESTATÍTICAS USANDO O R: Introdução à análise exploratória com gráficos (*ggplot*)**

**4.1 Primeiro passo: instalação e ativação do programa**

install.packages("ggplot2")

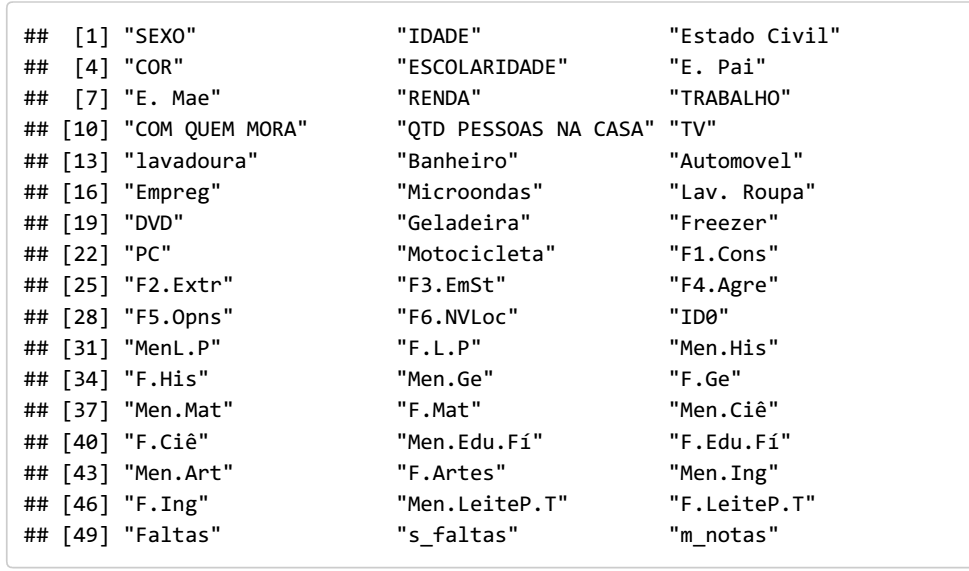
**library**(ggplot2)

**4.2 Abrir o banco de dados e explorar suas variáveis**

load("~/Dropbox/R Stat/senna.RData")

names(sennav1)

\*visualizar variáveis



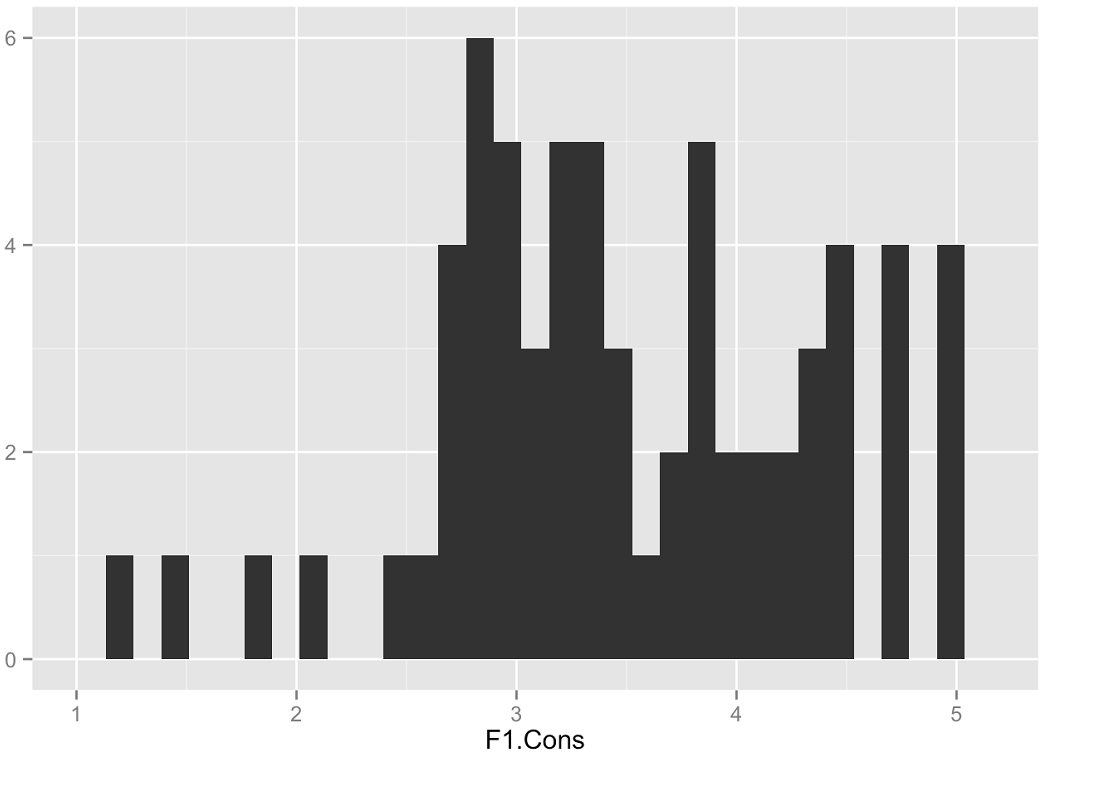
**4.3 Fazendo um histograma (análise de distribuição de uma variável)**

Comando:

ggplot(data=sennav1, aes(x=F1.Cons)) + geom\_histogram()

## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

Saída no programa:

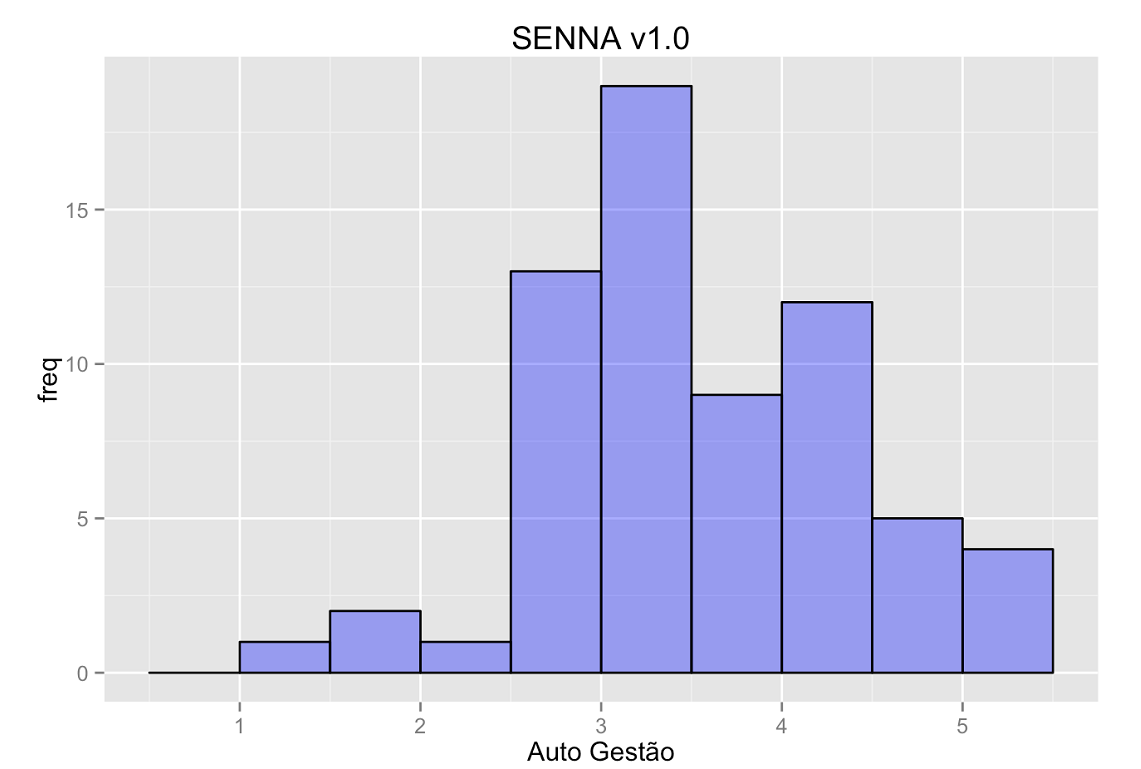


* 1. **Mudando detalhes dos gráficos (labs: títulos / alpha: transparência / binwidth: intervalo de x)**

Comando:

ggplot(data=sennav1, aes(x=F1.Cons)) + geom\_histogram(alpha =.4, fill ="blue", color="black", binwidth=.5) + labs(title="SENNA v1.0", x="Auto Gestão", y="freq")

Saída no programa:



**4.5 Comparando distribuições em função de uma segunda variável**

Passos/ direcionamentos:

* fill: cada valor da variável será mapeada em diferentes cores no preenchimento das barras
* variáveis ficam dentro de aes(), já as constantes ficam fora.

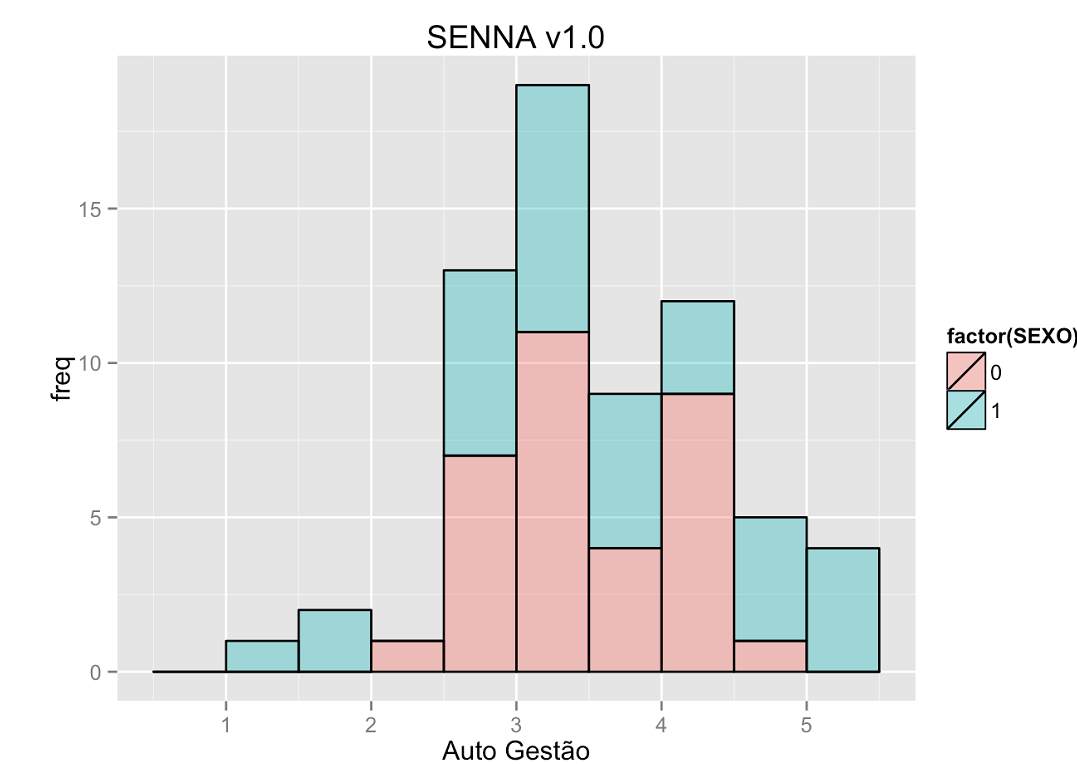
Comando:

ggplot(data=sennav1, aes(x=F1.Cons, fill = factor(SEXO))) +

geom\_histogram(alpha =.4, binwidth=.5, color = "black") +

labs(title="SENNA v1.0", x="Auto Gestão", y="freq")

Saída no programa:



**4.6 Comparando distribuições em dois gráficos separados**

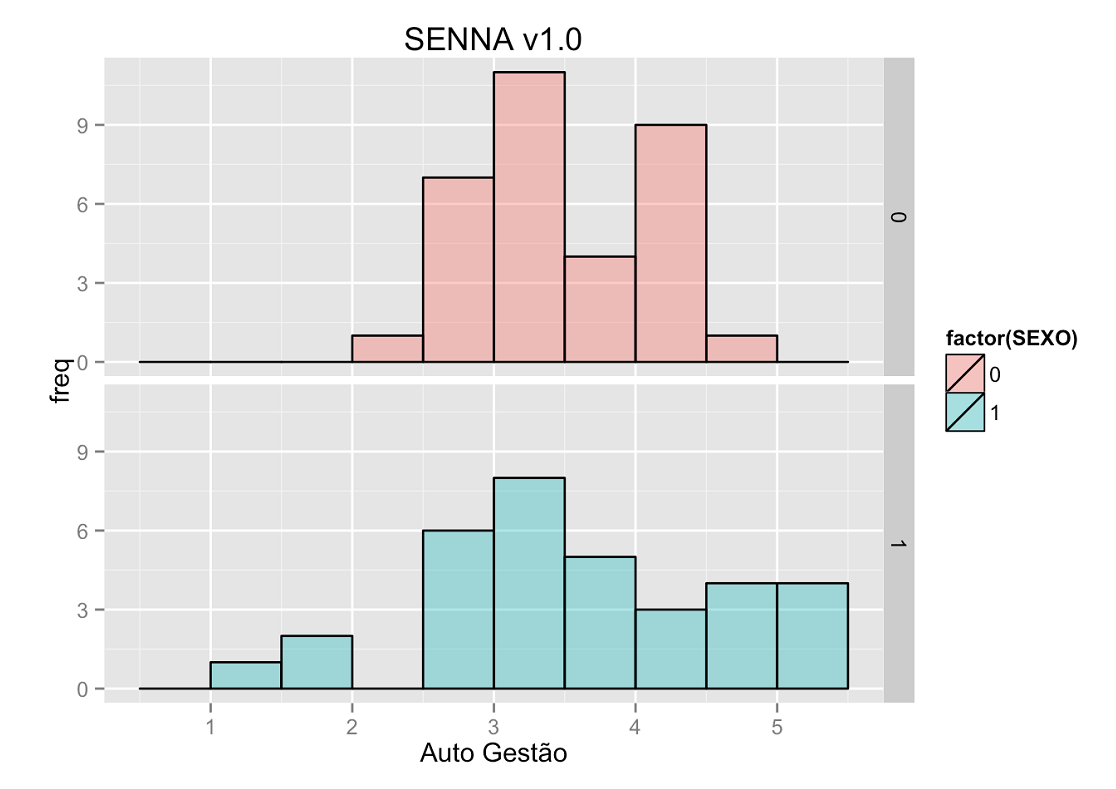
Passos/ direcionamentos:

* facet\_grid()
* dentro dos parêntesis coloque o nome da variável que indica os subgrupos assim: linha~coluna
* se quiser os gráficos um embaixo do outro: linha~.
* se quiser os gráficos um ao lado do: .~coluna

Comando:

* ggplot(data=sennav1, aes(x=F1.Cons, fill = factor(SEXO))) +
* geom\_histogram(alpha =.4, binwidth=.5, color = "black") +
* labs(title="SENNA v1.0", x="Auto Gestão", y="freq") +
* facet\_grid(SEXO~.)

Saída no programa:



**4.7 Gráfico densidade com tons de cinza**

Passos/ direcionamentos:

* theme\_bw()
* scale\_fill\_grey(start = 0, end = .9)

Comando:

ggplot(data=sennav1, aes(x=F1.Cons, fill = factor(ESCOLARIDADE))) +

geom\_density(alpha =.4) +

labs(title="SENNA v1.0", x="Auto Gestão", y="freq", fill = "ano") +

theme\_bw() + scale\_fill\_grey(start = 0, end = .9)

Saída no programa:



**4.8 Distribuição para diferentes subgrupos com Box plots**

Passos/ direcionamentos:

* note que agora dentro do aes(..) a variável y = F1.Cons e x = factor(ESCOLARIDADE)
* usamos fill = factor(ESCOLARIDADE) para pintar cada objeto com cores diferentes

Comando:

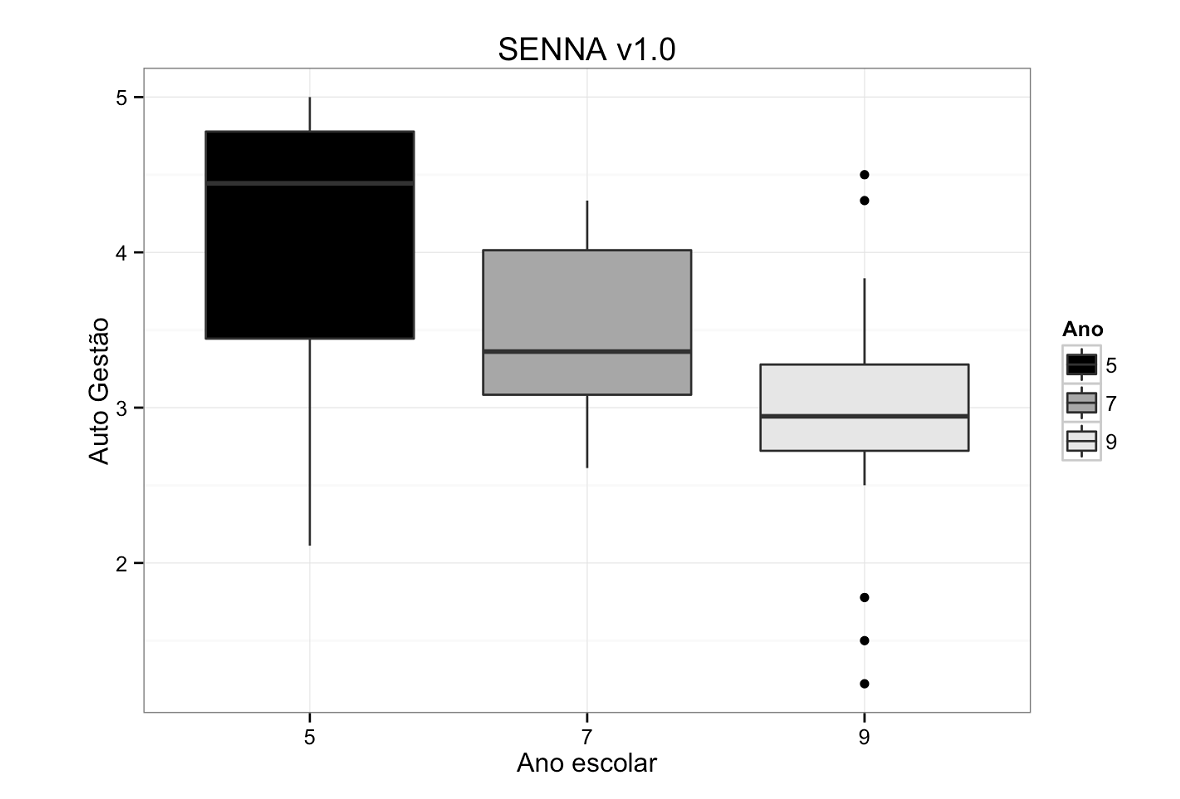
gplot(data=sennav1, aes(y=F1.Cons, x = factor(ESCOLARIDADE), fill = factor(ESCOLARIDADE))) +

geom\_boxplot() +

labs(title="SENNA v1.0", x="Ano escolar", y="Auto Gestão", fill = "Ano") +

theme\_bw() + scale\_fill\_grey(start = 0, end = .9):

Saída no programa:



**4.9 Correlação entre variáveis: gráfico de pontos *scaterplot***

Passos/ direcionamentos:

* + - geom\_point()
    - scale\_fill\_grey(start = 0, end = .9)

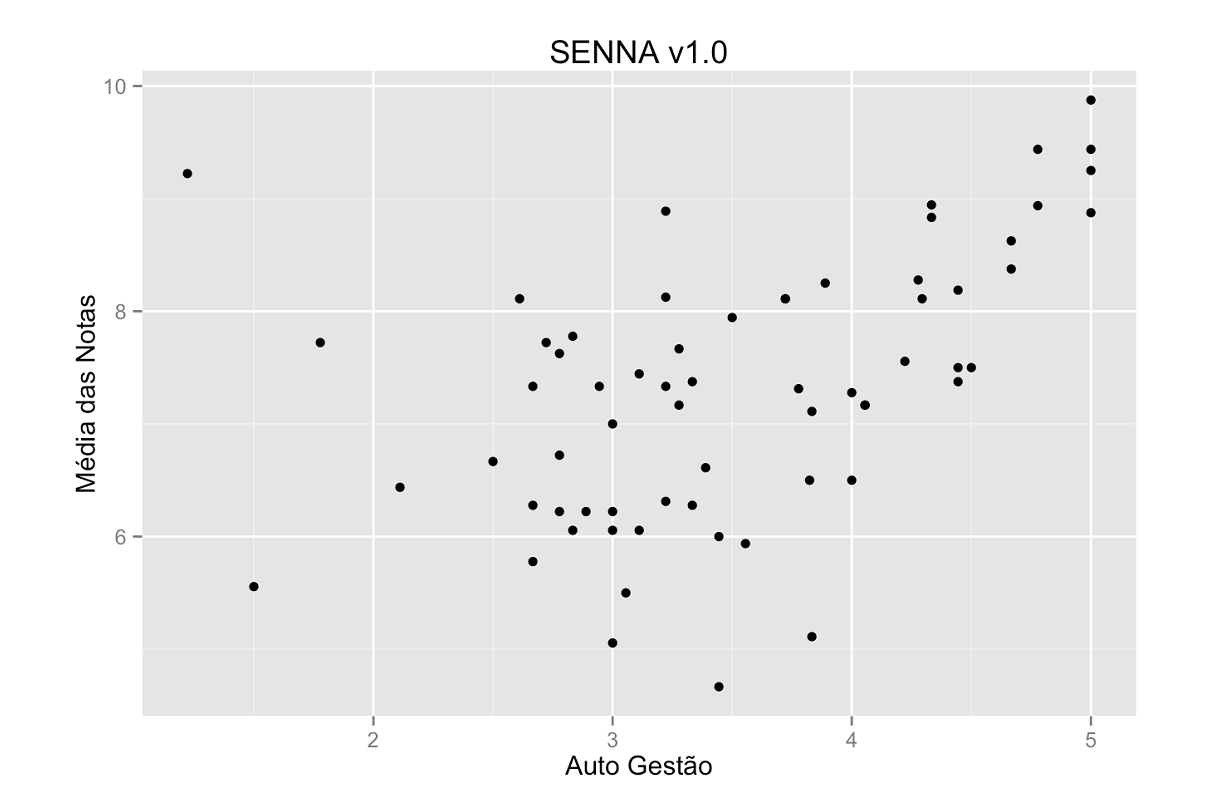
Comando:

ggplot(data=sennav1, aes(x=F1.Cons, y=m\_notas)) +

geom\_point() +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas")

Saída no programa:



**4.10 *Scaterplot* com reta de gregressão**

Passos/ direcionamentos:

* + - geom\_point()
    - scale\_fill\_grey(start = 0, end = .9)

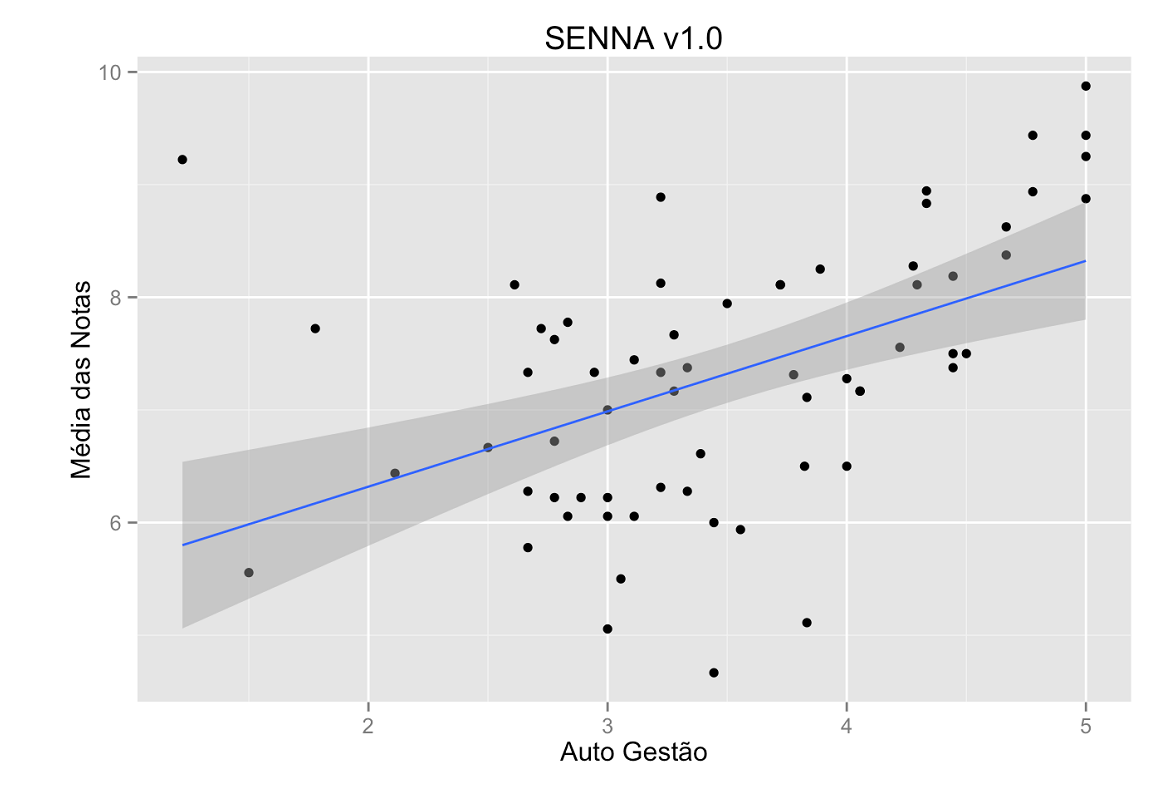
Comando:

ggplot(data=sennav1, aes(x=F1.Cons, y=m\_notas)) +

geom\_point() + geom\_smooth(method="lm") +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas")

Saída no programa:



**4.11 Correlação entre variáveis para diferentes subgrupos**

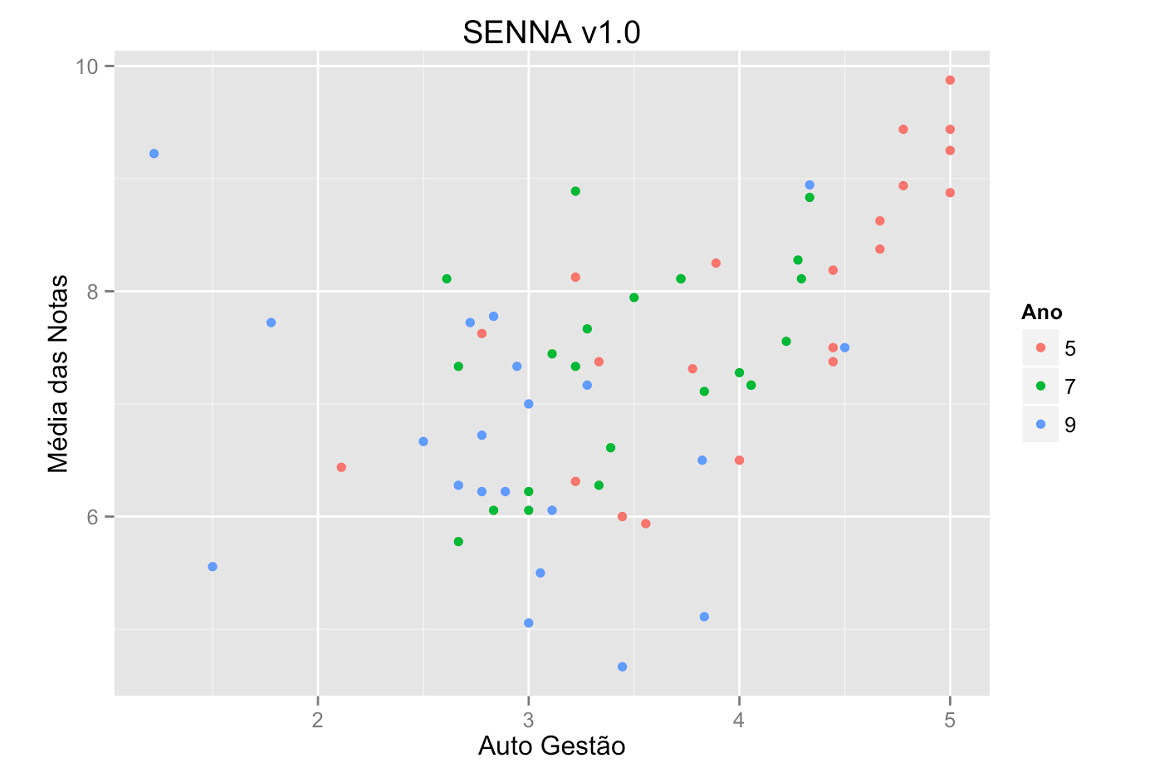
Comando:

ggplot(data=sennav1, aes(x=F1.Cons, y=m\_notas, color = factor(ESCOLARIDADE))) +

geom\_point() +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas", color= "Ano")

Saída no programa:



**4.12 Correlação entre variáveis para diferentes subgrupos adicionando retas de regressão**

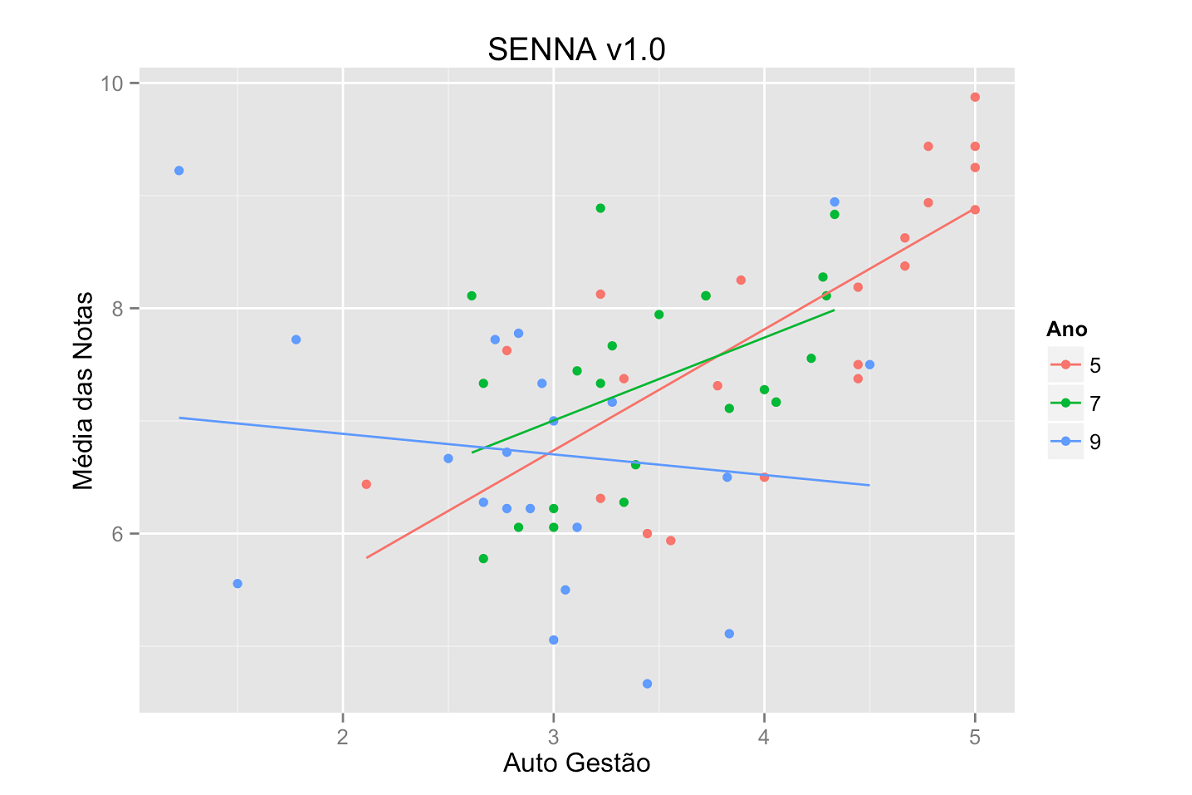
Comando:

ggplot(data=sennav1, aes(x=F1.Cons, y=m\_notas, color = factor(ESCOLARIDADE))) +

geom\_point() + geom\_smooth(method="lm", se=FALSE) +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas", color= "Ano")

Saída no programa:



**4.13 Modificando a aparência do gráfico**

Passos/ direcionamentos:

* + - Adição da forma dos pontos
    - Note que agora usei scale\_color\_grey()

Comando:

ggplot(data=sennav1, aes(x=F1.Cons, y=m\_notas, color = factor(ESCOLARIDADE), shape = factor(ESCOLARIDADE))) +

geom\_point() + geom\_smooth(method="lm", se=FALSE) +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas", color= "Ano", shape = "Ano") +

theme\_bw() + scale\_color\_grey(start = .2, end = .9)

Saída no programa:

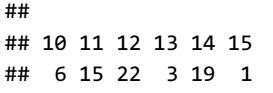


* 1. Descrevendo os dados tabelas de frequência

Comando:

table(sennav1$IDADE)

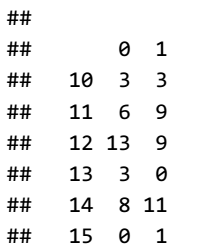
Saída no programa:



Comando:

table(sennav1$IDADE, sennav1$SEXO)

Saída no programa:



Comando:

table(sennav1$IDADE, sennav1$ESCOLARIDADE)

Saída no programa:



**4.15 Descrevendo os dados com estatísticas descritivas**

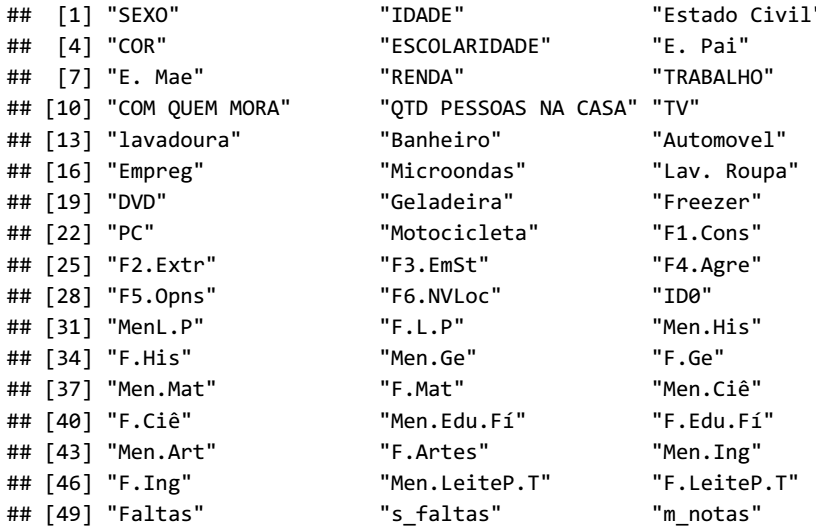
Passos/ direcionamentos:

* + - média, moda e desvio padrão
    - assimetria e curtose

Comando:

names(sennav1)

Saída no programa:

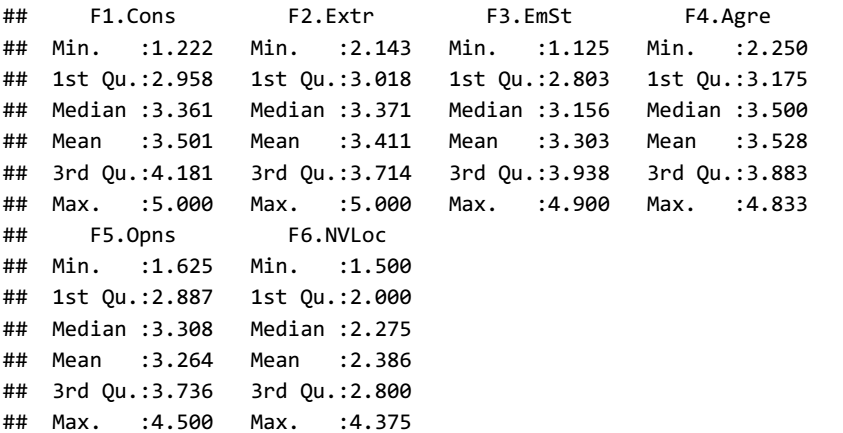


Comando:

summary(sennav1[ , 24:29 ])

\*os números são as variáveis selecionadas a serem descritas

Saída no programa:

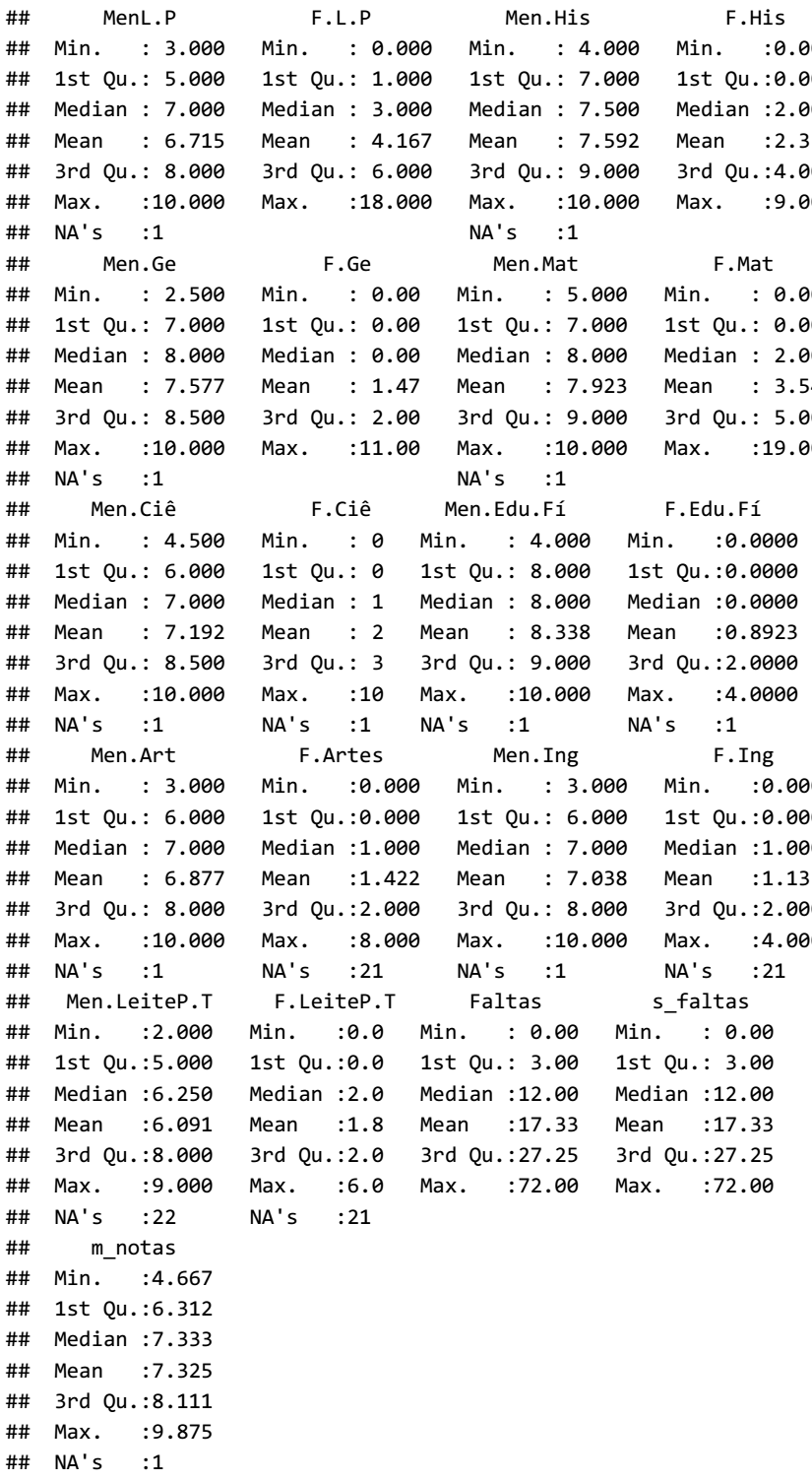


Comando:

summary(sennav1[ , 31:51 ])

\*os números são as variáveis selecionadas a serem descritas

Saída no programa:



Comando:

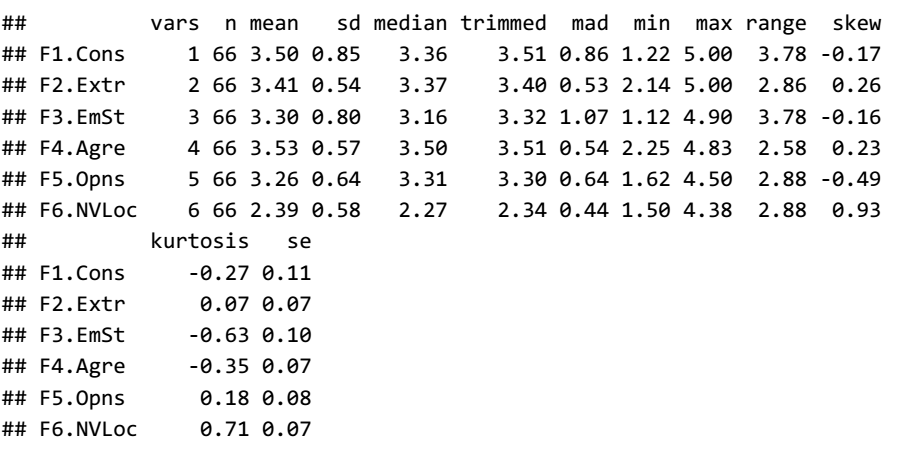
**library**(psych)

\*ativar esse pacote para rodar as demais análises

Comando:

escribe(sennav1[ , 24:29 ])

Saída no programa:

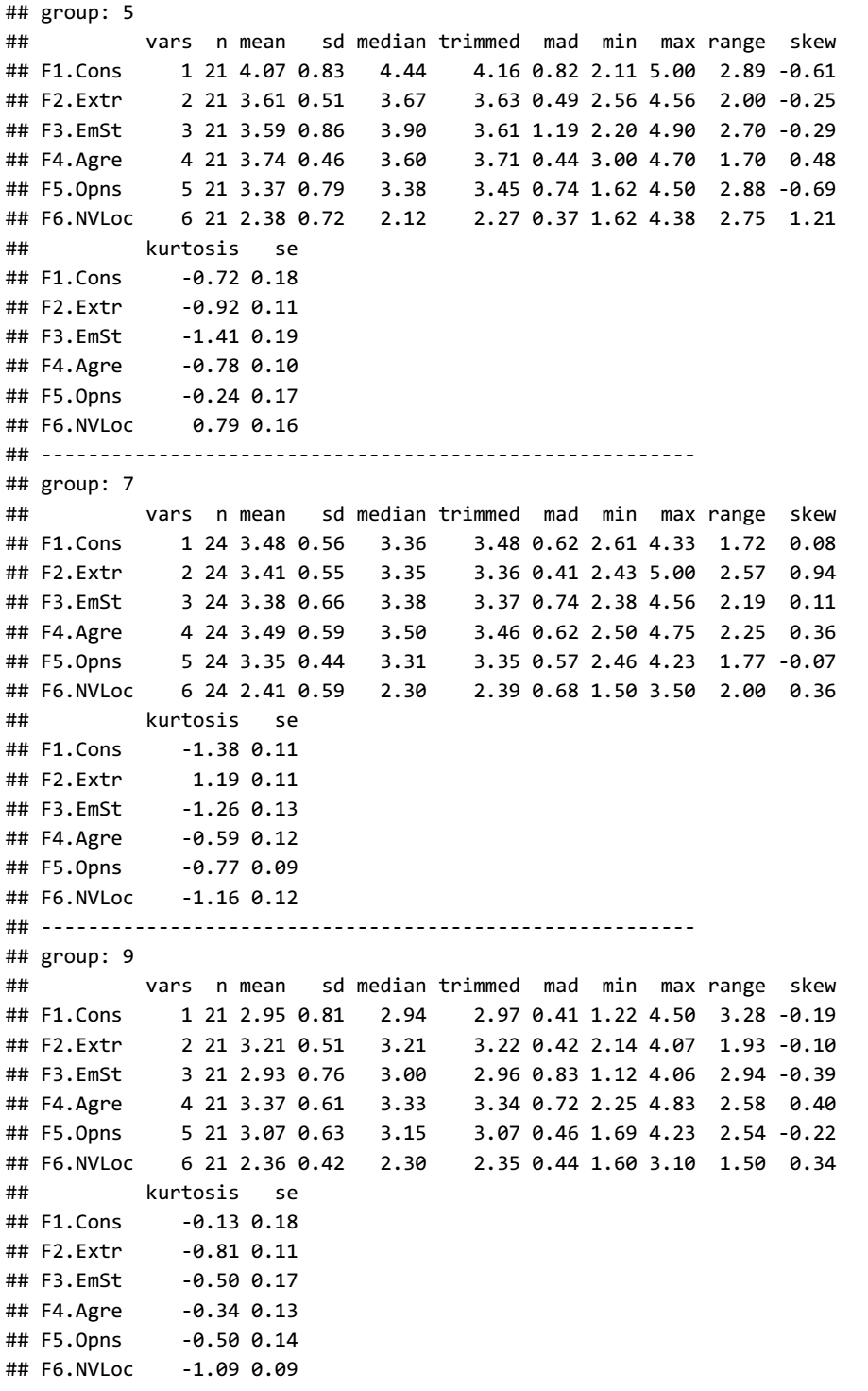


Comando:

describeBy(sennav1[ , 24:29 ], group=sennav1$ESCOLARIDADE)

\*análises de variáveis por grupos

Saída no programa:

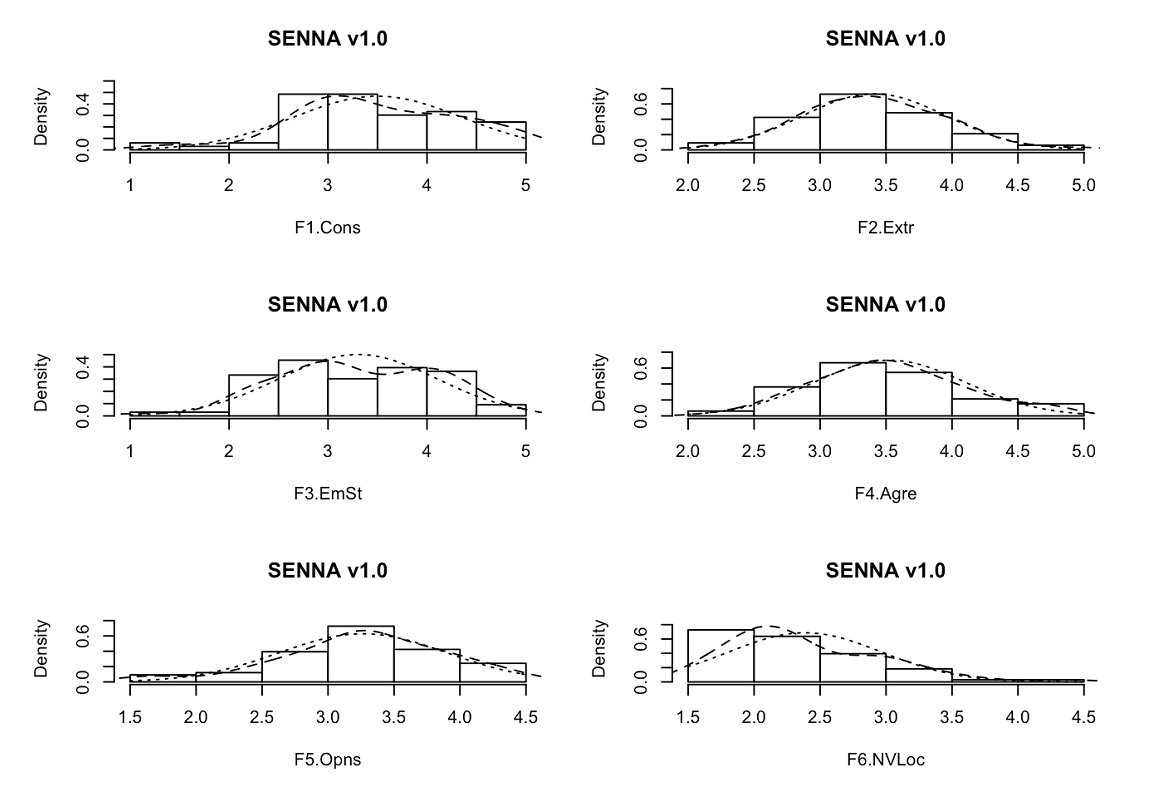


Comando:

multi.hist(sennav1[ , 24:29 ], main = "SENNA v1.0")

\*histogramas das variáveis

Saída no programa:

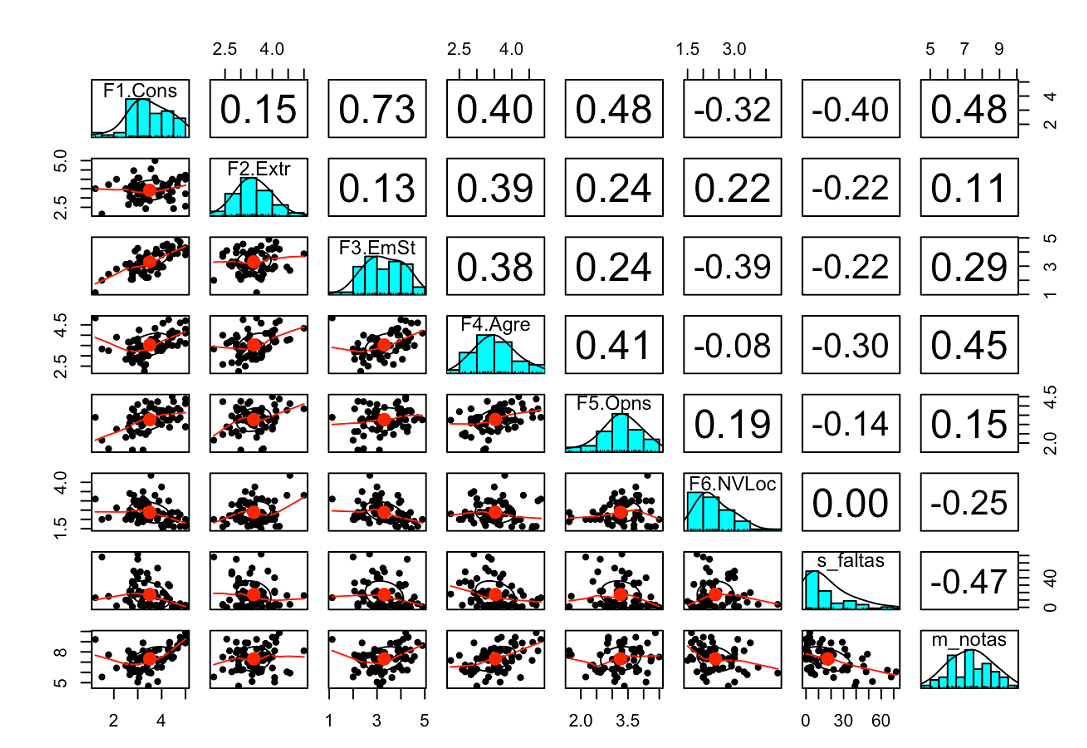


* 1. Análise de Correlação

Comando:

pairs.panels(sennav1[ , c( 24:29, 50:51) ])

Saída no programa:

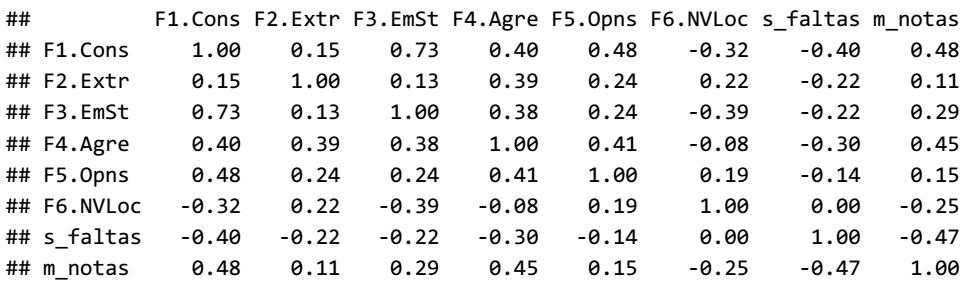


Comando:

r <- cor(sennav1[ , c( 24:29, 50:51)] ,use="pairwise")

round(r, digits = 2)

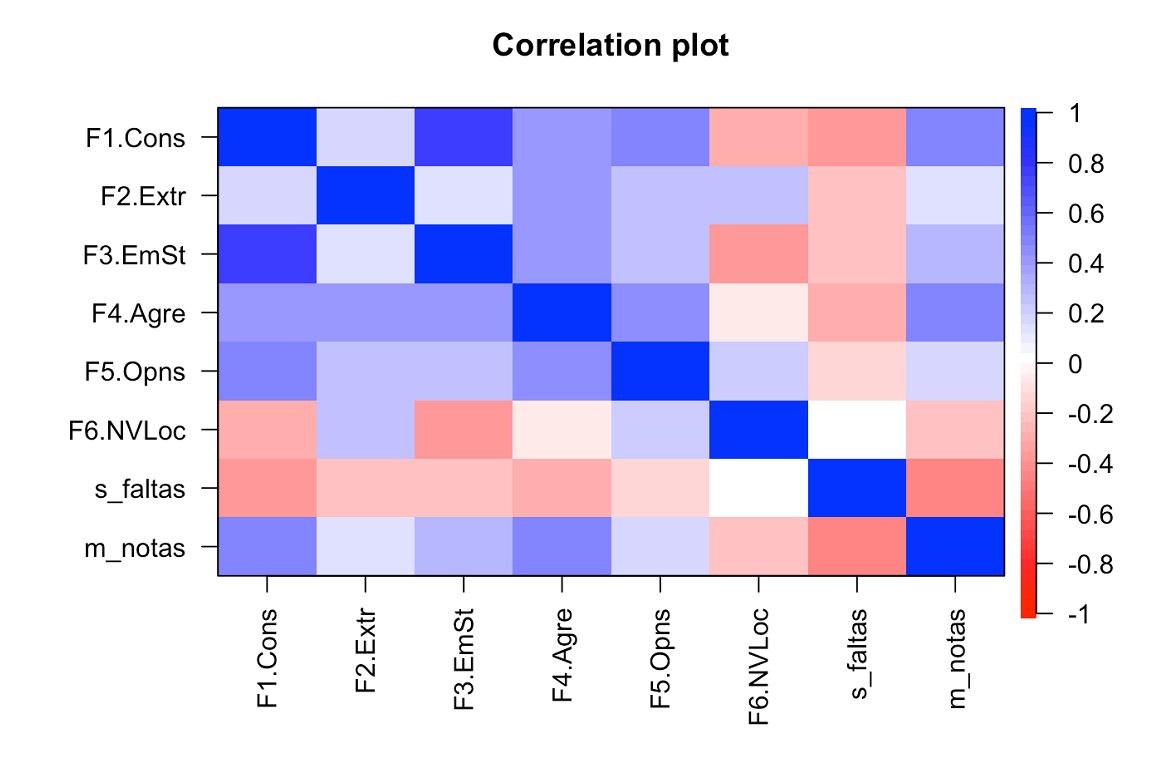
Saída no programa:



Comando:

cor.plot(cor(sennav1[ , c(24:29, 50:51)], use="pair"))

Saída no programa:



## Exercício 3: aprendendo estrutura de dados

* Examine as distribuições dos escores entre duas variáveis do seu banco de dados
* Examine as correlações entre fatores de um instrumento do seu banco de dados com a frequência
* Examine se as correlações entre esses fatores variam por gênero

**5. ANÁLISES INFERENCIAIS SIMPLES: t-teste, ANOVA e regressão simples.**

Comando:

setwd("~/Dropbox/R Stat")

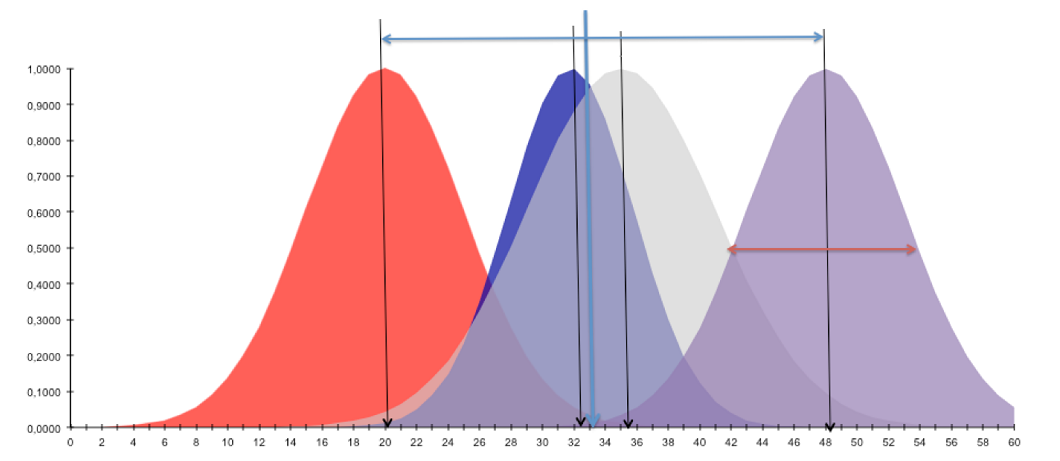
\*sempre lembrar de abrir o banco de dados e os arquivos habilitados a serem trabalhados

**5.1 Análise de variância simples (ANOVA)**

Fundamentos

* Comparação de médias n grupos
* Pergunta de pesquisa: há diferenças entre as médidas ? Há uma tendência global as pessoas em um dos grupos terem notas mais altas ou mais baixas que outro grupo?
* Os n grupos são amostras de uma distribuição única ou de diferentes distribuições?





#### **5.2 Prática 1: Há diferenças em auto-gestão entre alunos de séries diferentes?**

Comando:

*# Abrir banco de dados*

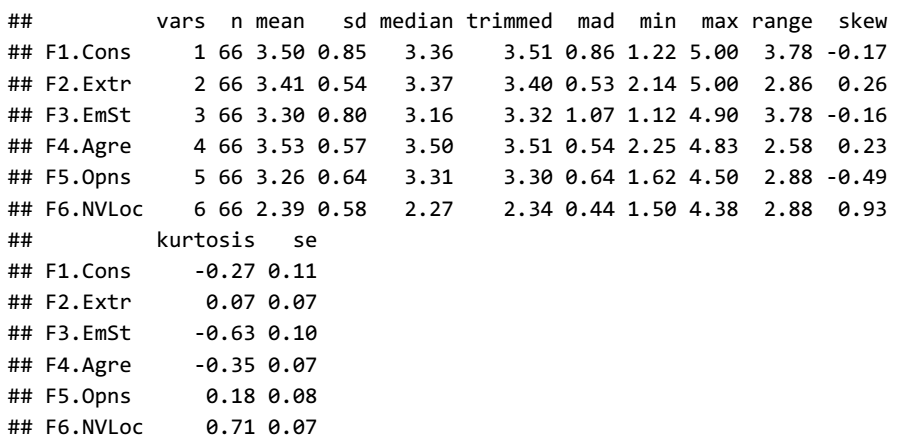
load("senna.RData")

*# Análise descritivas*

**library**(psych)

describe(sennav1[ , 24:29])

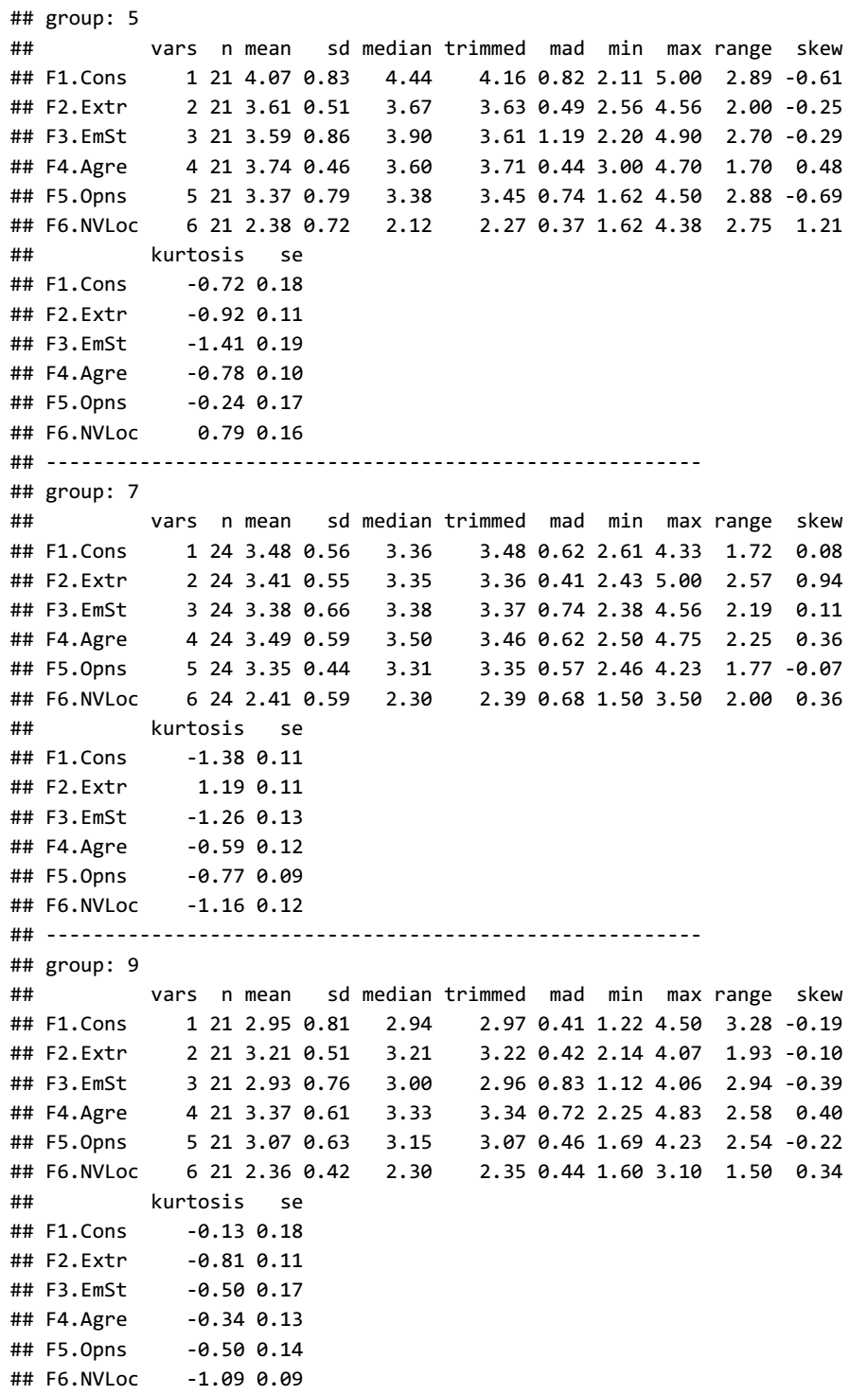
Saída no programa:



Comando:

describeBy(sennav1[ , 24:29], group = sennav1$ESCOLARIDADE)

Saída no programa:



Comando:

*# Cria escolaridade como uma variável "factor"*

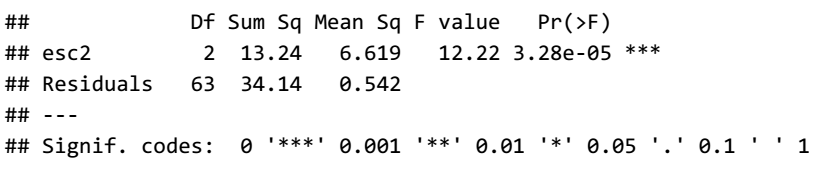
sennav1$esc2 <- factor(sennav1$ESCOLARIDADE)

*# ANOVA VD: auto gestão VI: escolaridade*

fit <- aov(F1.Cons ~ esc2, data = sennav1)

summary(fit)

Saída no programa:

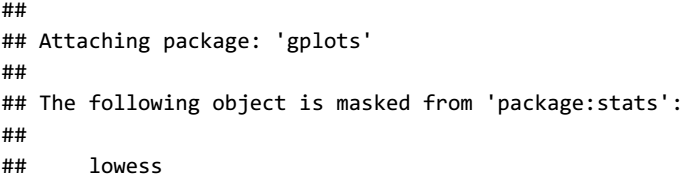


Comando:

*# Figura comparando as médias*

**library**(gplots)

Saída no programa:

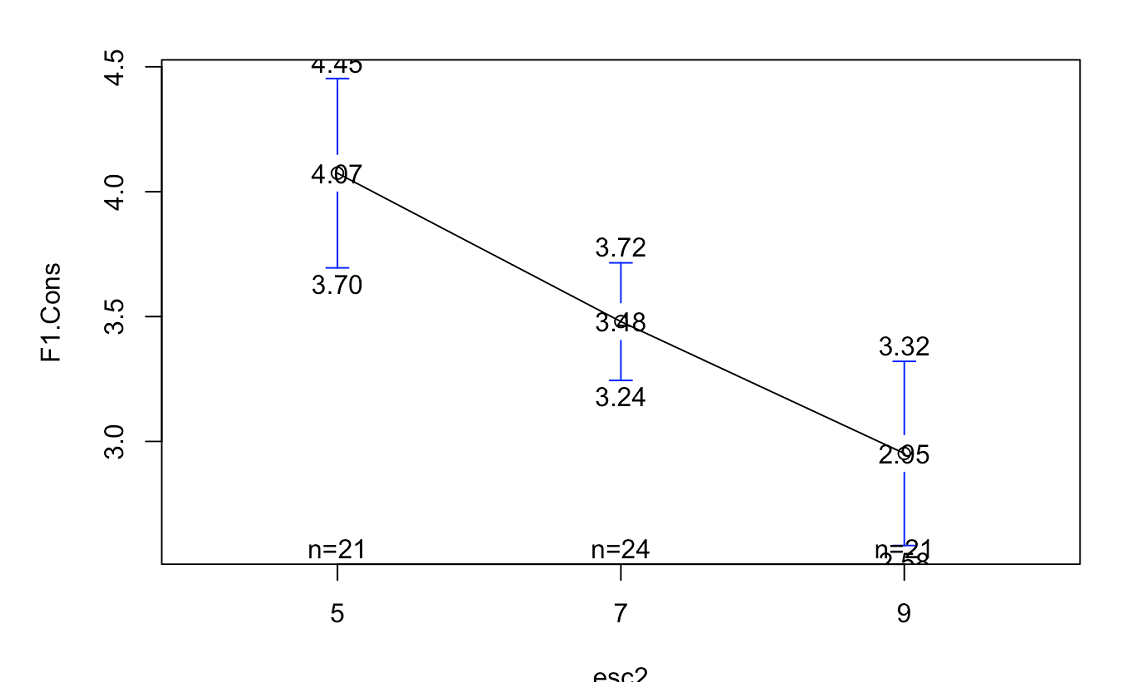


Comando:

plotmeans(F1.Cons ~ esc2, data = sennav1, bars =TRUE, ci.label = TRUE,

mean.labels=TRUE, digits = 2)

Saída no programa:

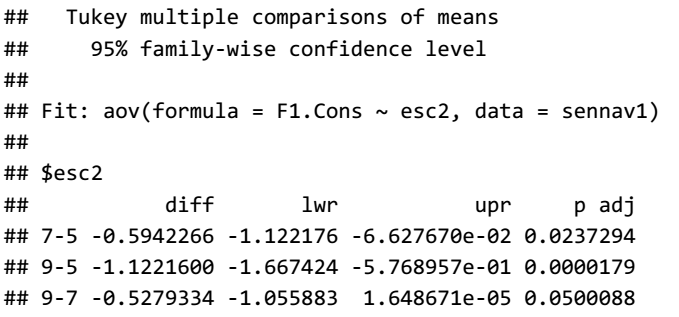


Comando:

*# Comparações post-hoc*

TukeyHSD(fit)

Saída no programa:



#### **5.3 t-teste - Diferenças entre gêneros**

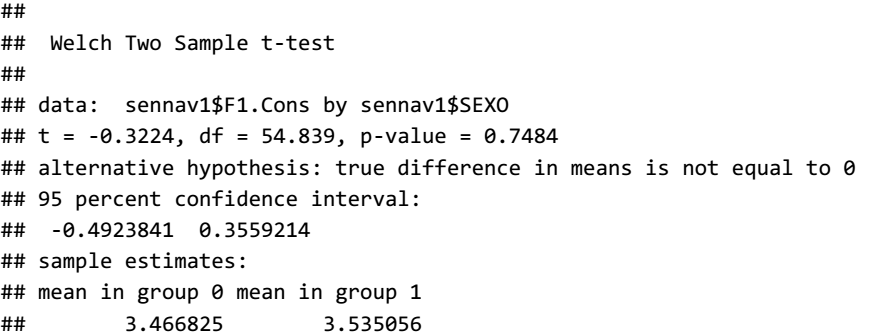
Comando:

*# Cria escolaridade como uma variável "factor"*

sennav1$esc2 <- factor(sennav1$ESCOLARIDADE)

t.test(sennav1$F1.Cons~sennav1$SEXO)

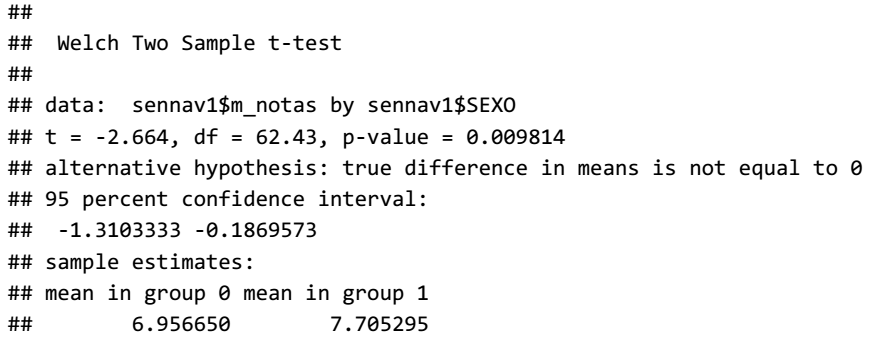
Saída no programa:



Comando:

t.test(sennav1$m\_notas~sennav1$SEXO)

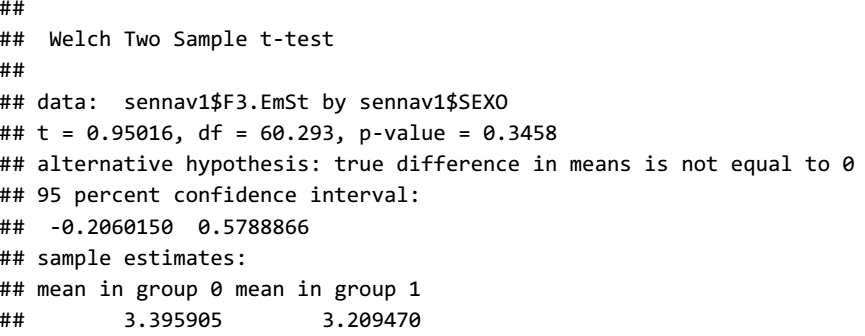
Saída no programa:



Comando:

t.test(sennav1$F3.EmSt~sennav1$SEXO)

Saída no programa:



**5.4 Análise de Regressão**

\* Para regressão tenho que transformar a variável em escore Z.

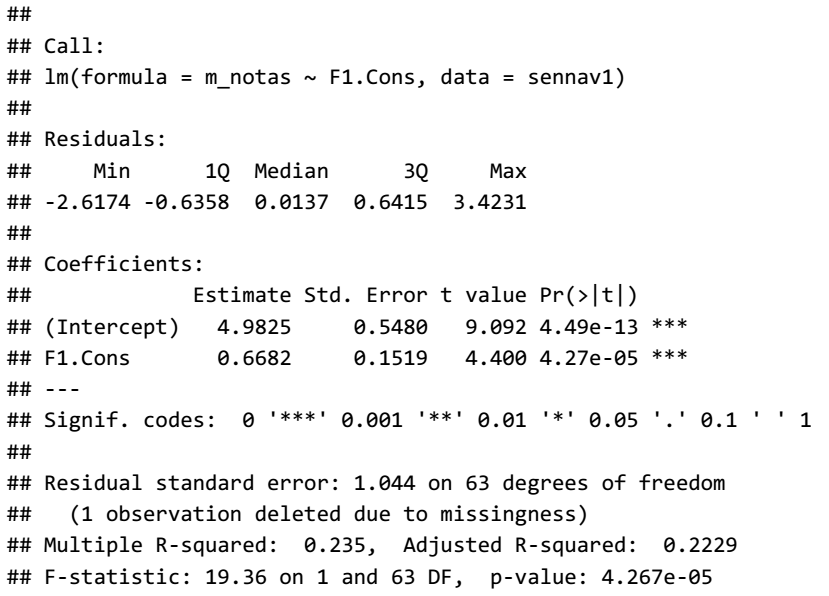
Comando:

*# Cria escolaridade como uma variável "factor"*

fit2 <- lm( m\_notas ~ F1.Cons , data=sennav1)

summary(fit2) *# show results*

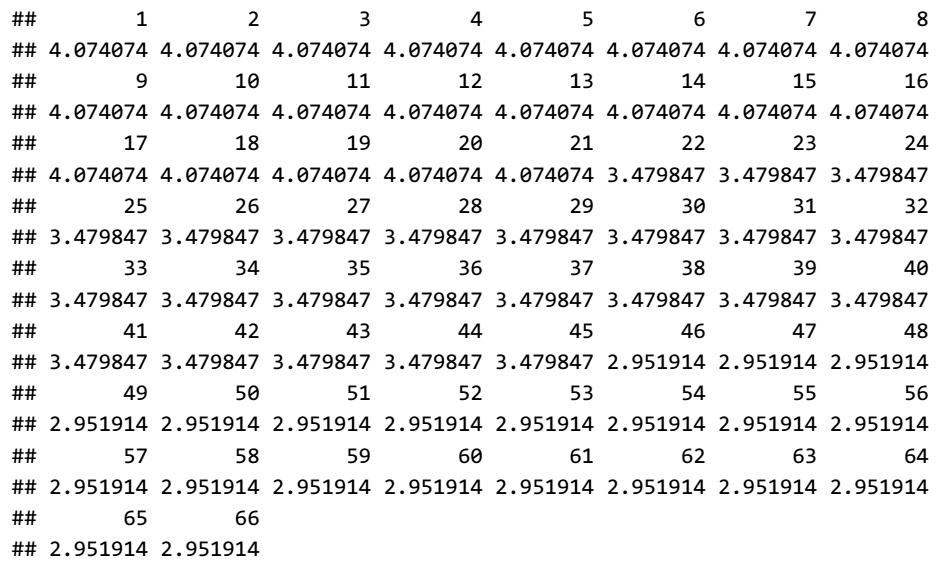
Saída no programa:



Comando:

fitted(fit) *# predicted values*

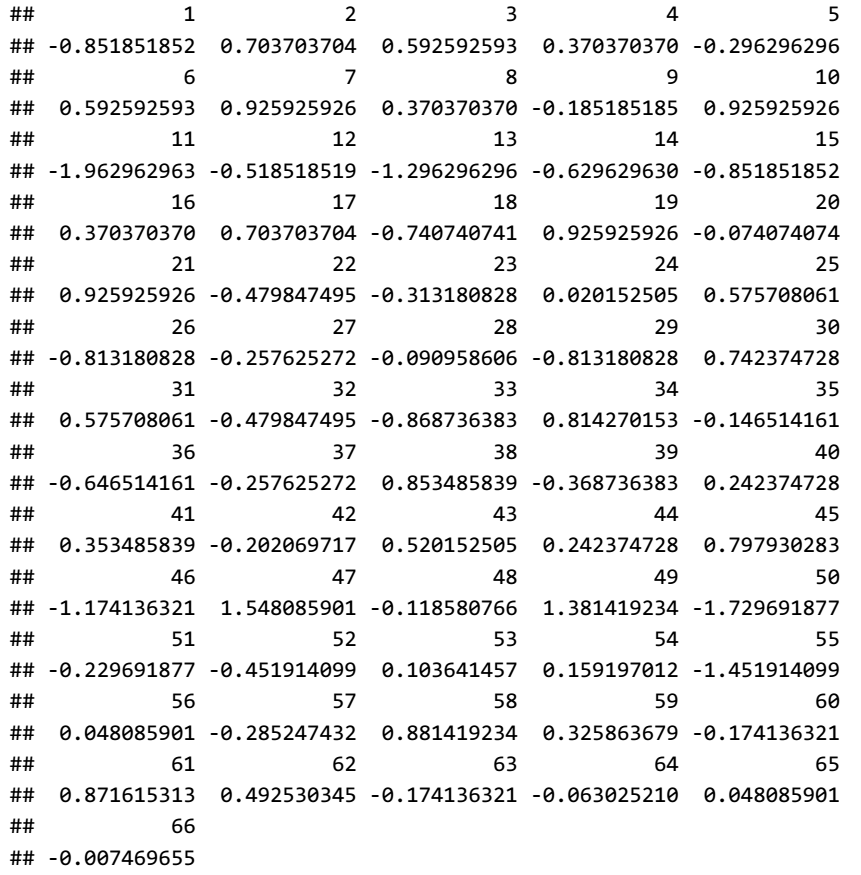
Saída no programa:



Comando:

residuals(fit) *# residuals*

Saída no programa:

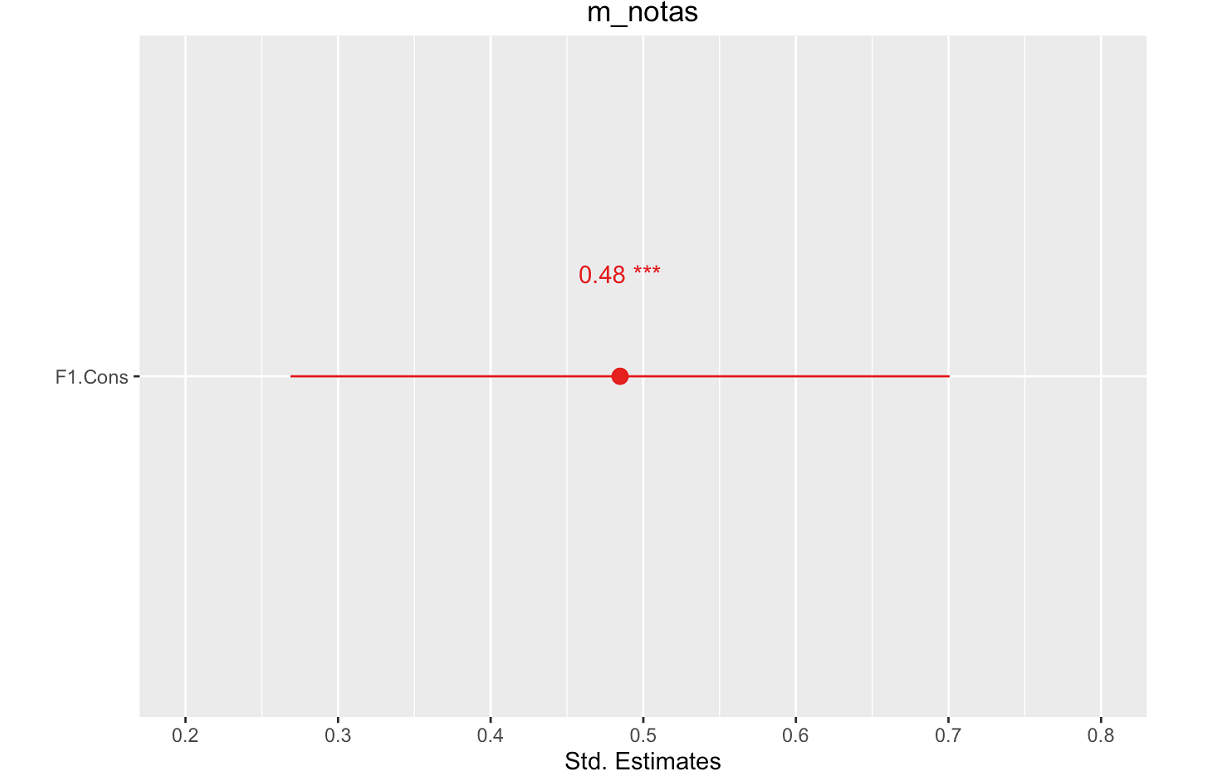


Comando:

**library**(sjPlot)

sjp.lm(fit2, type = "std")

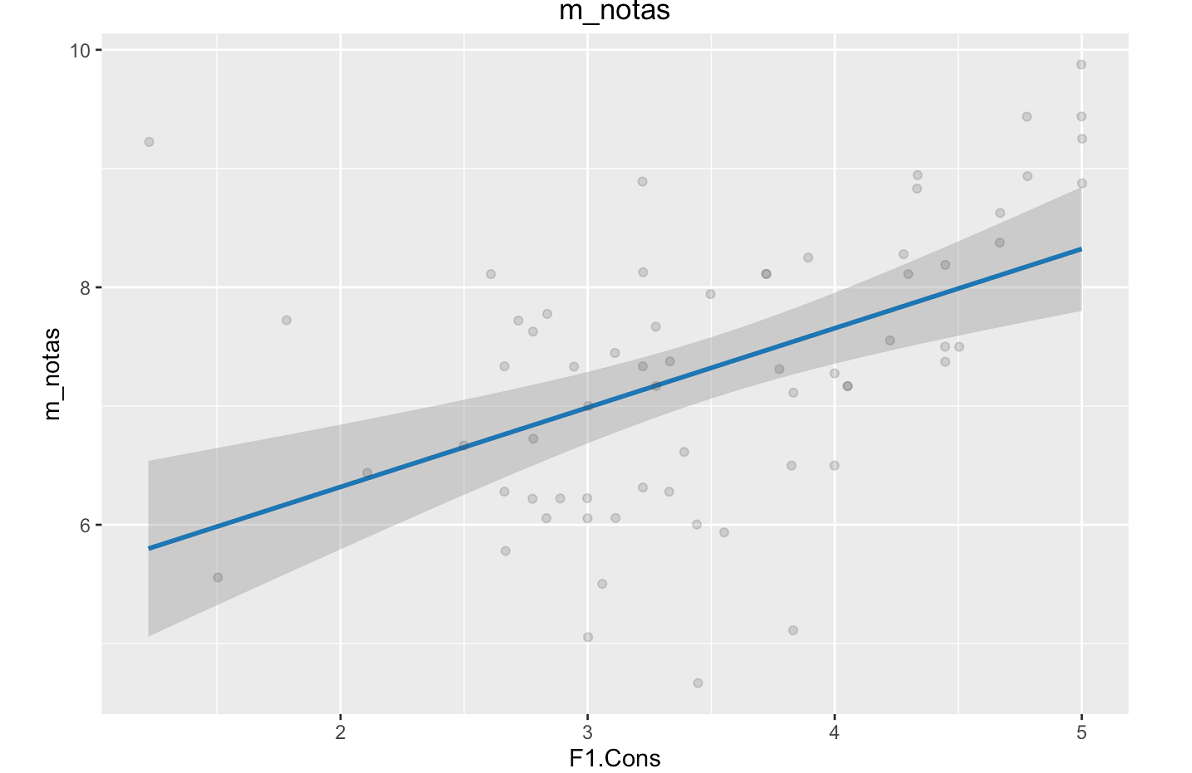
Saída no programa:

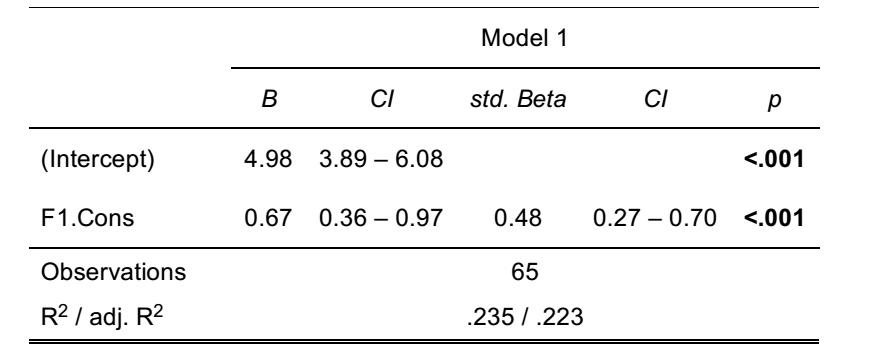


Comando:

sjp.lm(fit2, type = "pred")

Saída no programa:





### **Exercício 3**

* Faça a ANOVA considerando os fatores restantes do seu banco de dados
* Faça a regressão predizendo faltas a partir dos fatores do seu instrumento utilizado

**5. ANÁLISES INFERENCIAIS AVANÇADAS: ANOVA fatorial, ANCOVA e regressão múltipla.**

setwd("~/Dropbox/R Stat")

### **Análise de variância fatorial**

* Comparação de médias n grupos em delineamentos fatoriais com mais de uma variável independente ou de grupo
* Pergunta de pesquisa: há efeitos principais ? Há efeitos de interação entre as variáveis interativos (o efeito de uma VI muda nos subníveis da outra)

#### **Prática com o senna v1 (two way ANOVA): Há diferenças em auto gestão entre alunos de séries diferentes ? Essa diferença é a mesma entre homes e mulheres? ANOVA 3 X 2 (Escolaridade X Sexo)**

Comando:

*# Abrir banco de dados*

load("senna.RData")

*# Cria escolaridade e sexo como uma variável "factor"*

sennav1$esc2 <- factor(sennav1$ESCOLARIDADE)

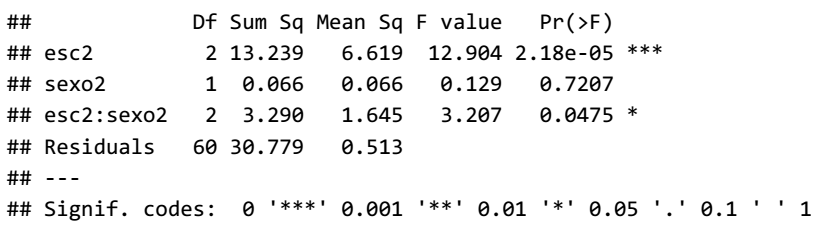
sennav1$sexo2 <- factor(sennav1$SEXO)

*# ANOVA VD: auto gestão VI's: escolaridade e sexo*

fit <- aov(F1.Cons ~ esc2\*sexo2, data = sennav1)

summary(fit)

Saída no programa:

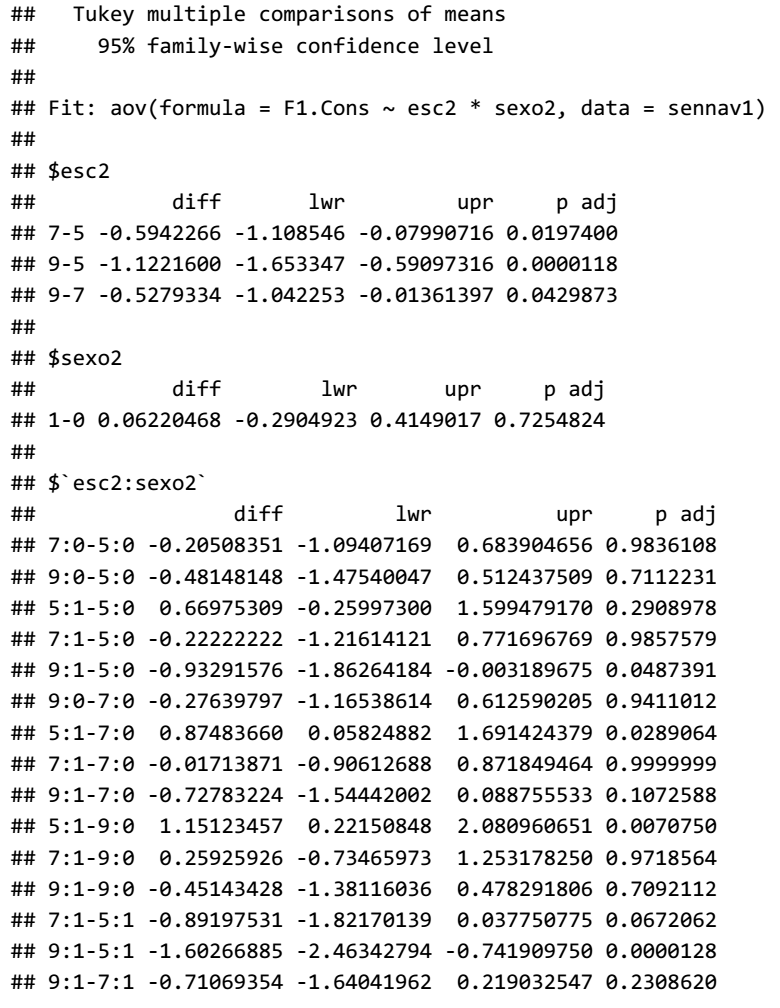


Comando:

*# Comparações post-hoc*

TukeyHSD(fit)

Saída no programa:

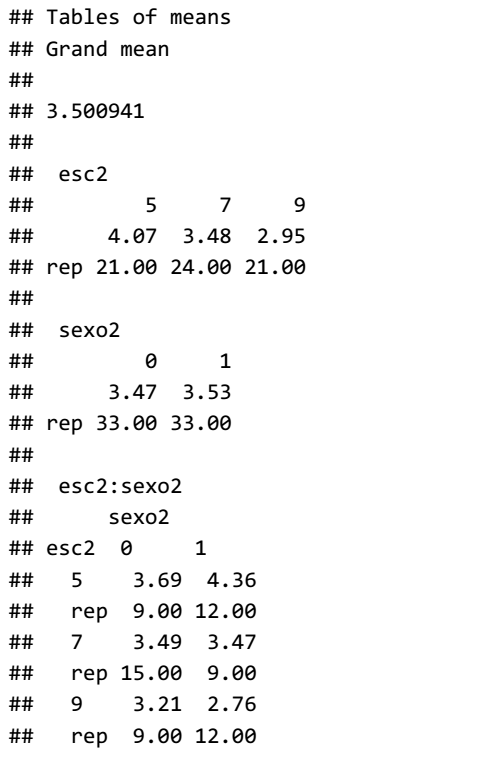


Comando:

*# Alguns programas em: http://www.personality-project.org/r/r.anova.html*

print(model.tables(fit,"means"),digits=3)

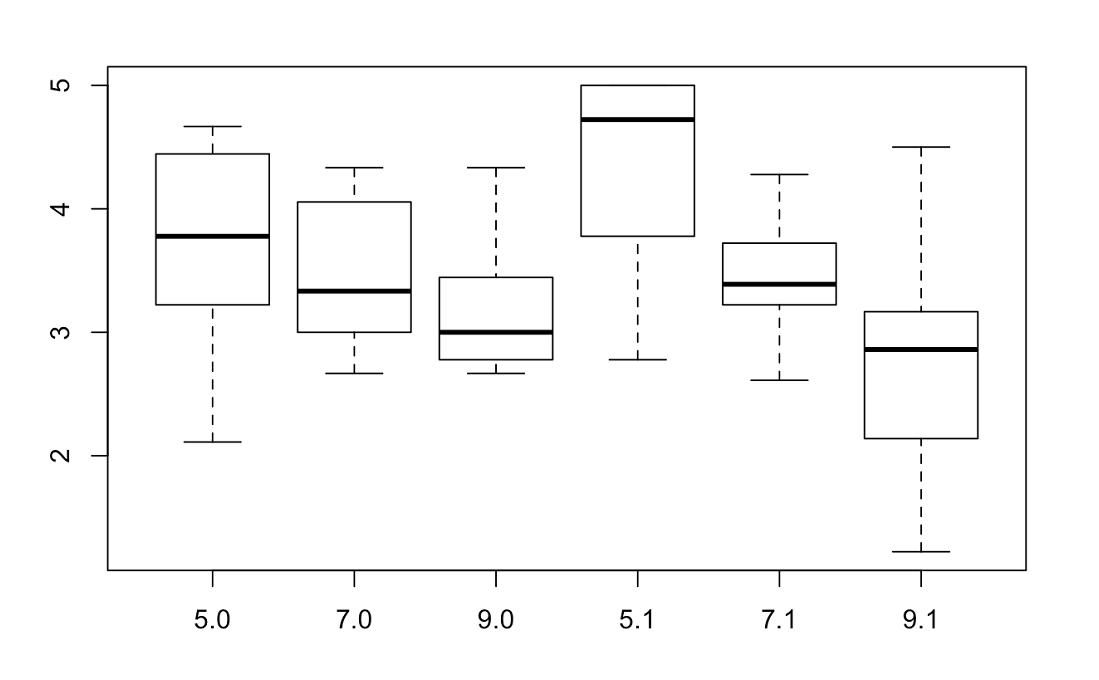
Saída no programa:



Comando:

boxplot(F1.Cons ~ esc2\*sexo2,data=sennav1 )

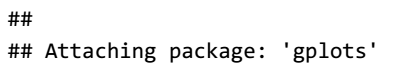
Saída no programa:

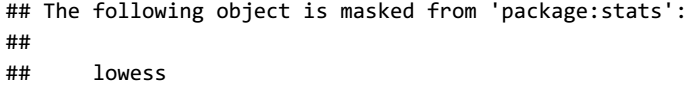


Comando:

**library**(gplots)

Saída no programa:





Comando:

plotmeans(F1.Cons ~ interaction(esc2, sexo2, sep=" "),

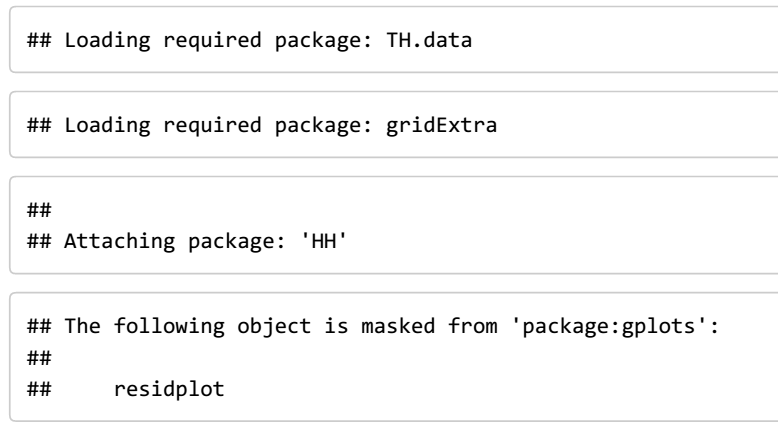
connect=list(c(1,2,3),c(4,5,6)),

data = sennav1)

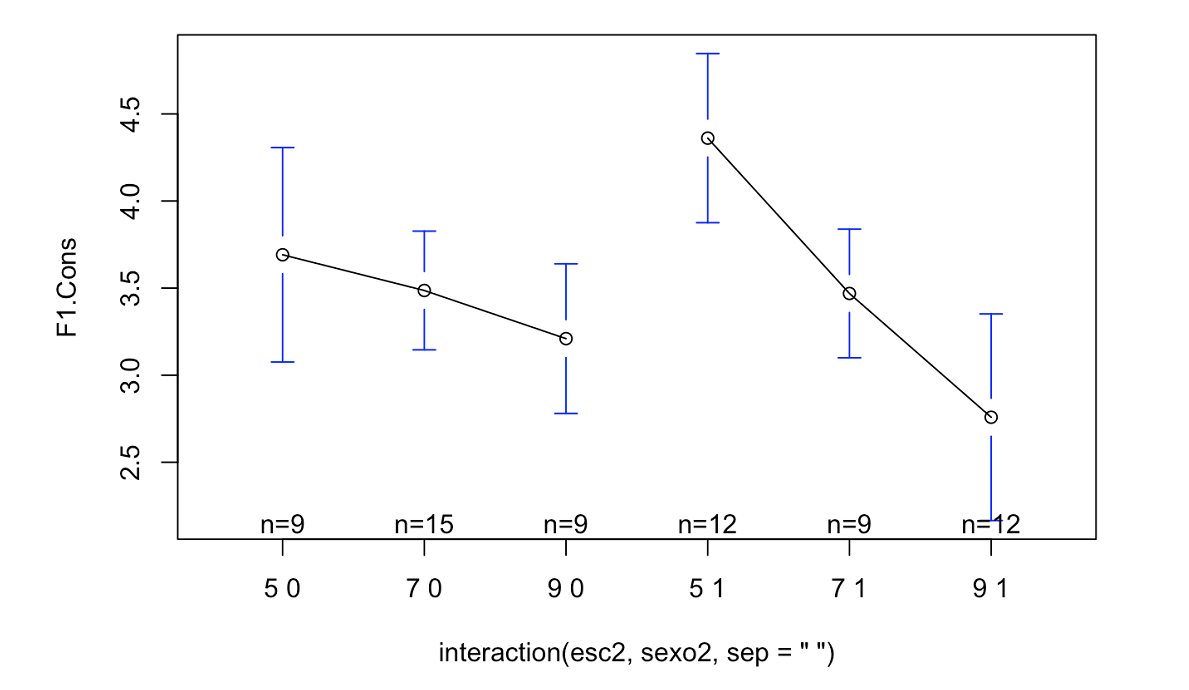
**library**(HH)

Saída no programa:





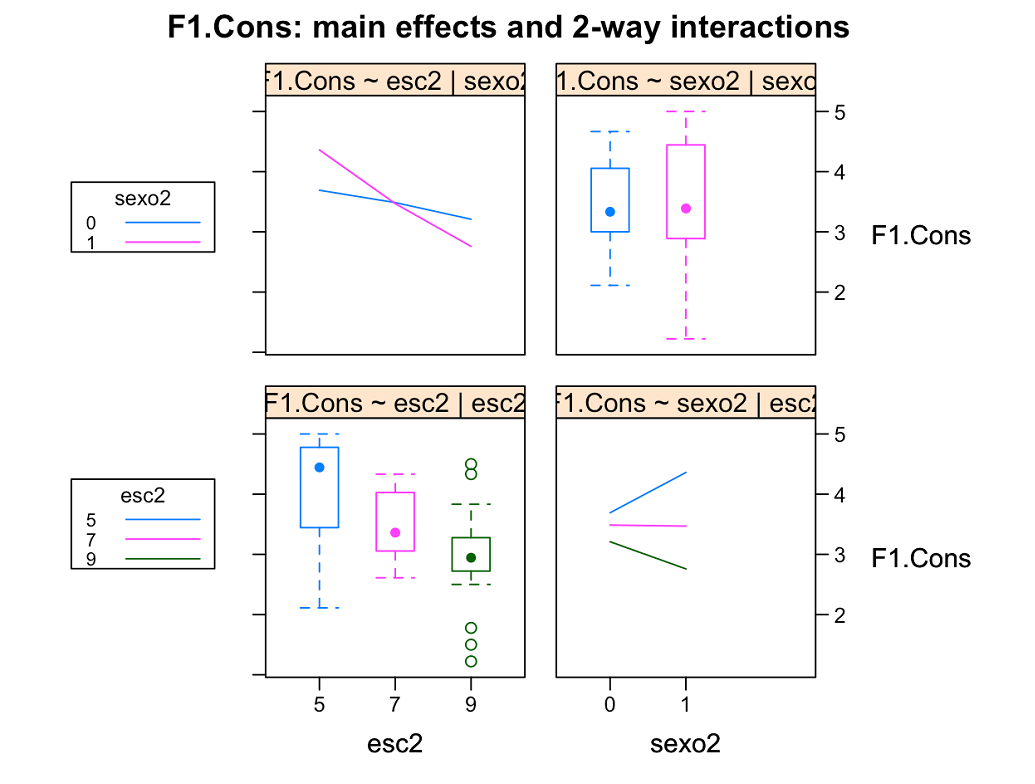
Saída no programa:



Comando:

interaction2wt(F1.Cons ~ esc2\*sexo2, data = sennav1)

Saída no programa:



#### **Prática com o senna v1 (two way ANOVA): Há diferenças em auto-gestão entre alunos de séries diferentes? Essa diferença é a mesma entre alunos com desempenho acima ou abaixo da mediana? ANOVA 3 X 2 (Escolaridade X Desempenho)**

Comando:

*# Cria uma divisão pela mediana*

sennav1$m\_notas2 <-cut(sennav1$m\_notas,

breaks =c(0, median(sennav1$m\_notas, na.rm =TRUE), 10))

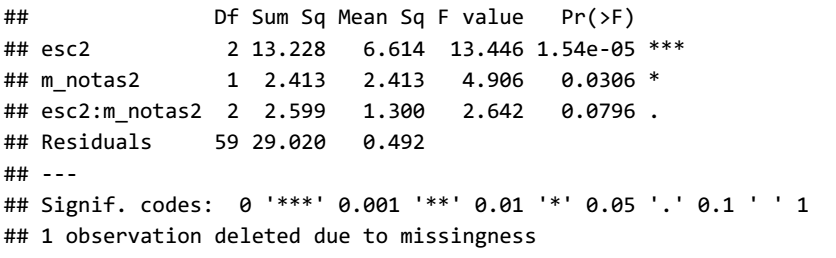
sennav1$sexo2 <- factor(sennav1$SEXO)

*# ANOVA VD: auto gestão VI's: escolaridade e sexo*

fit <- aov(F1.Cons ~ esc2\*m\_notas2, data = sennav1)

summary(fit)

Saída no programa:

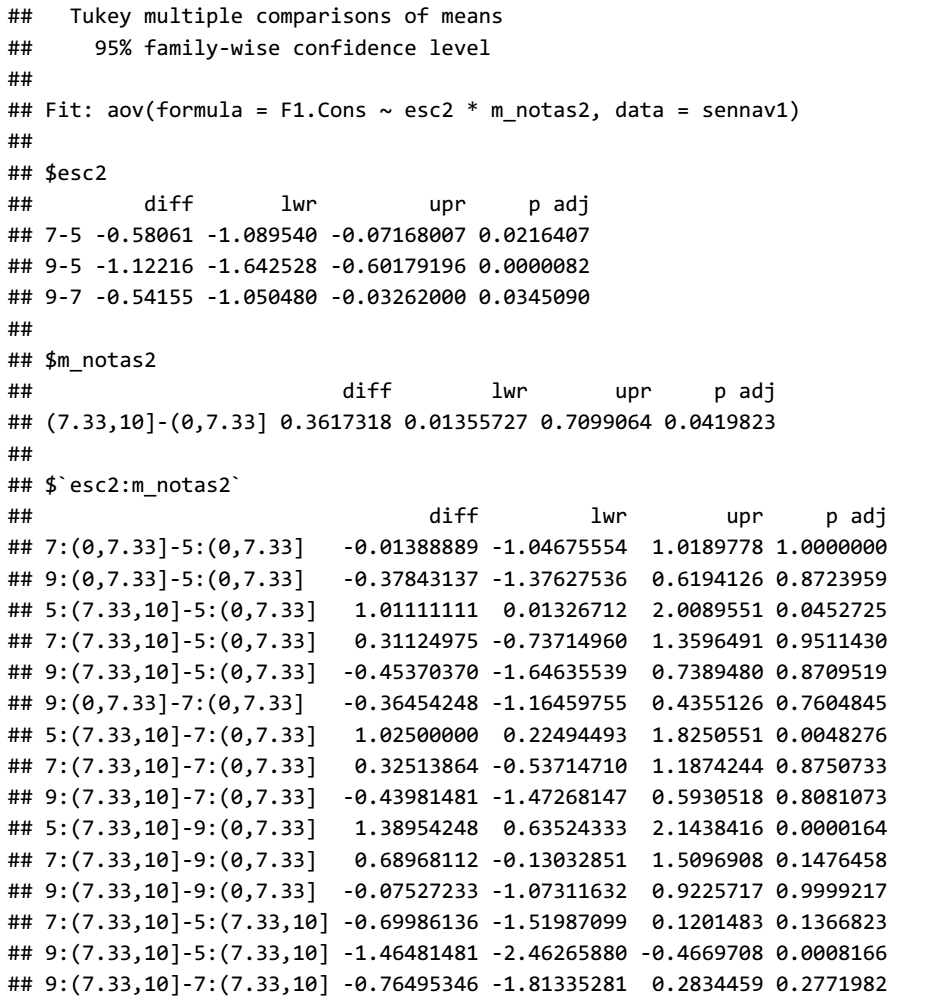


Comando:

*# Comparações post-hoc*

TukeyHSD(fit)

Saída no programa:

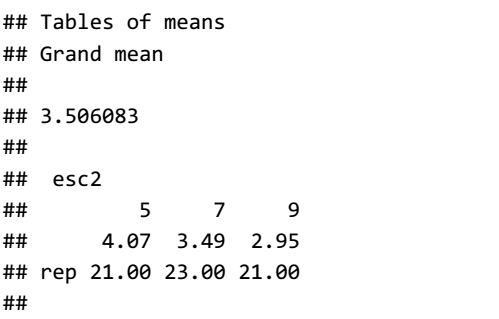


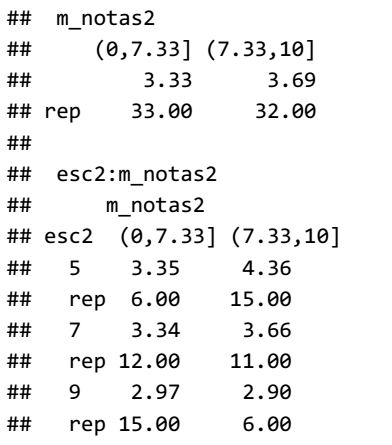
Comando:

*# Figuras*

print(model.tables(fit,"means"),digits=3)

Saída no programa:

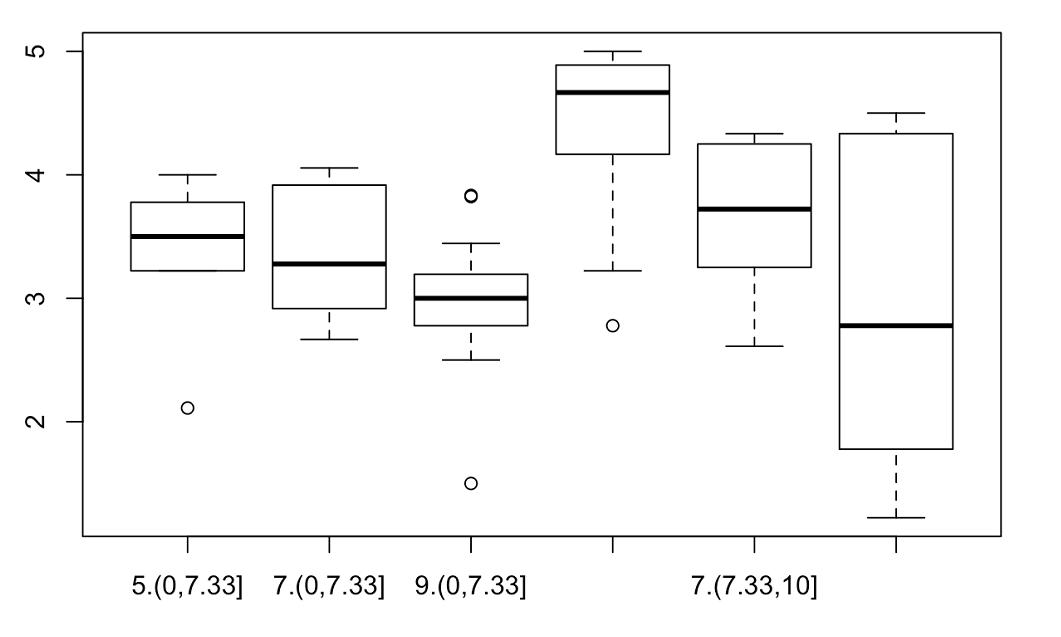




Comando:

boxplot(F1.Cons ~ esc2\*m\_notas2,data=sennav1)

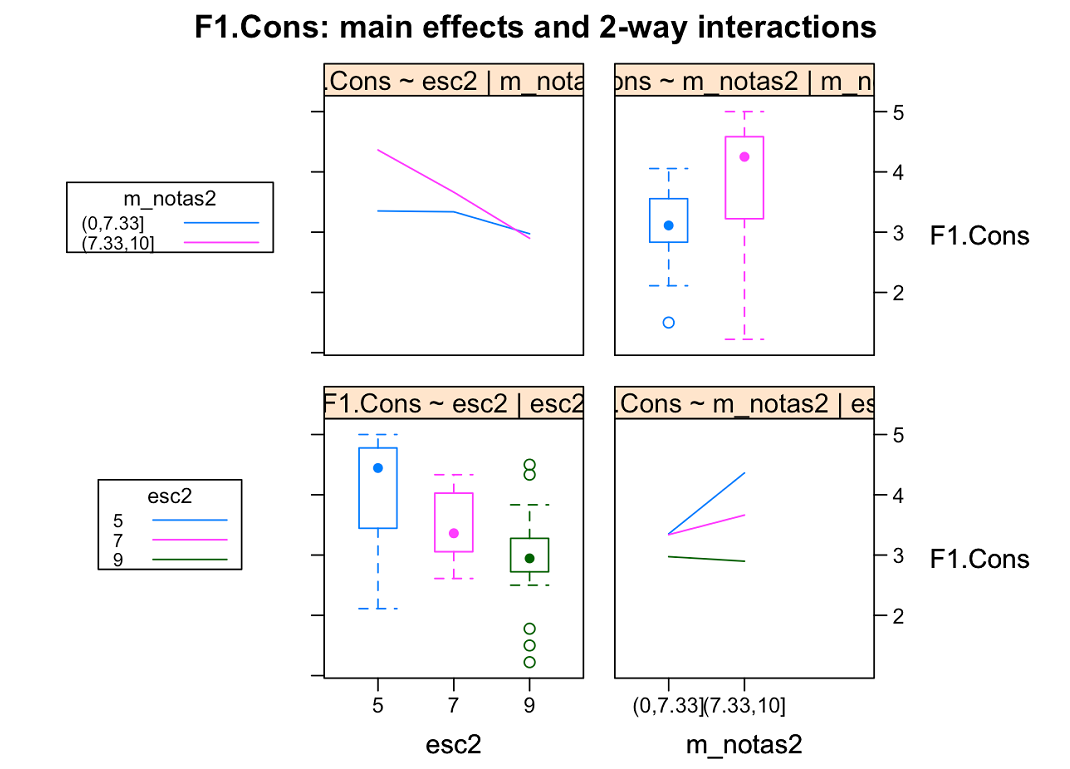
Saída no programa:



Comando:

interaction2wt(F1.Cons ~ esc2\*m\_notas2, data = sennav1)

Saída no programa:



#### **Análise de regressão múltipla? Qual contribuição única para predizer notas das variáveis socioemocionais ?**

Comando:

*# Correlação entre notas e habilidades socioemocionais*

**library**(sjPlot)

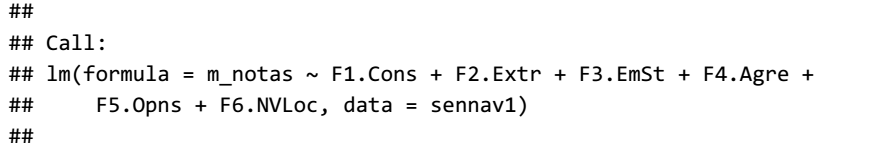
sjt.corr(sennav1[ , c("s\_faltas", "m\_notas", "F1.Cons", "F2.Extr", "F3.EmSt", "F4.Agre", "F5.Opns", "F6.NVLoc")] , triangle = "lower")

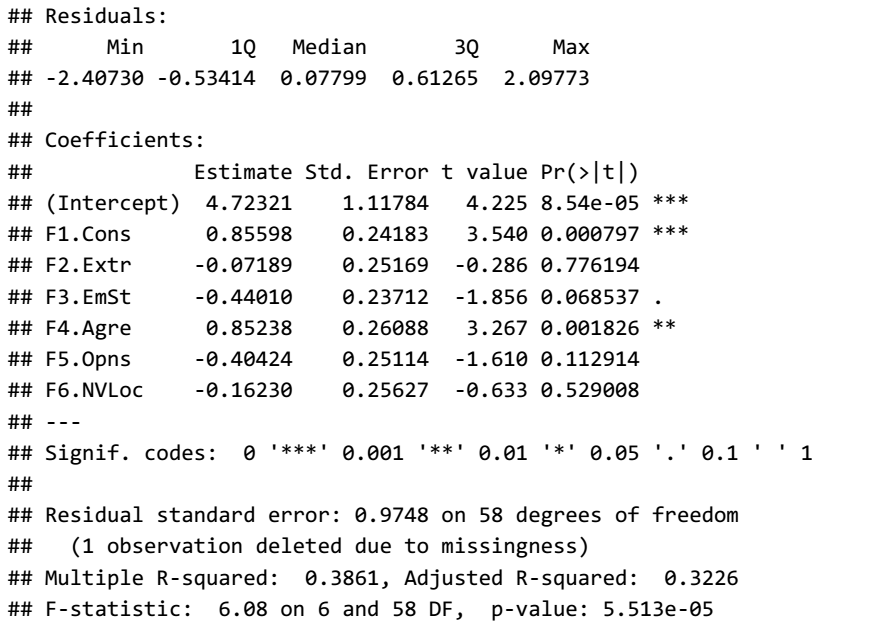
*# Regressão múltipla VD=Notas e VI's variáveis do Senna*

fit2 <- lm( m\_notas~F1.Cons+F2.Extr+F3.EmSt+F4.Agre+F5.Opns+F6.NVLoc, data=sennav1)

summary(fit2)

Saída no programa:





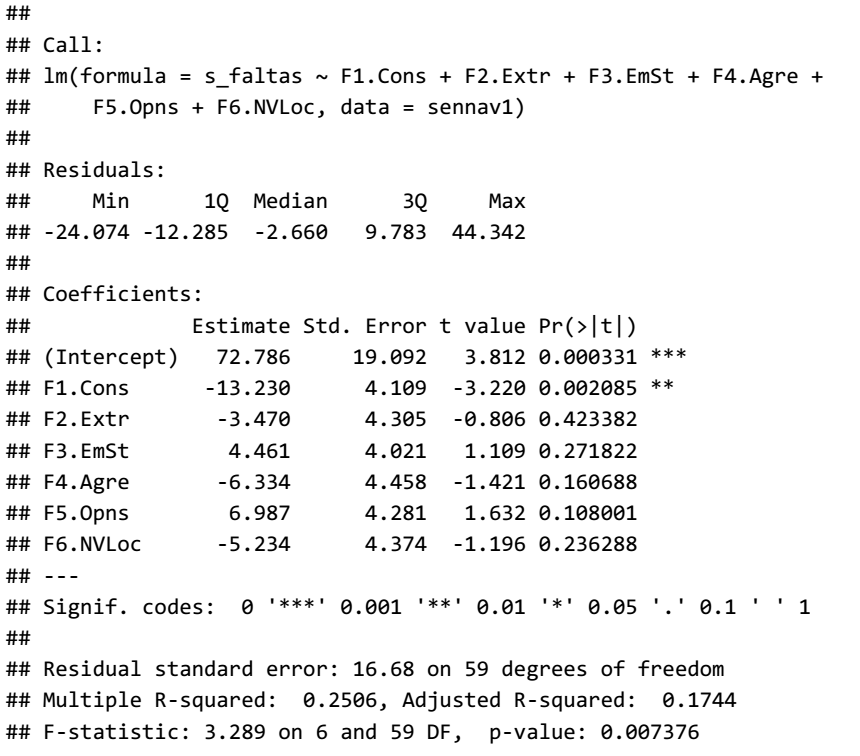
Comando:

*# Regressão múltipla VD=Frequência e VI's variáveis do Senna*

fit3 <- lm( s\_faltas~F1.Cons+F2.Extr+F3.EmSt+F4.Agre+F5.Opns+F6.NVLoc, data=sennav1)

summary(fit3)

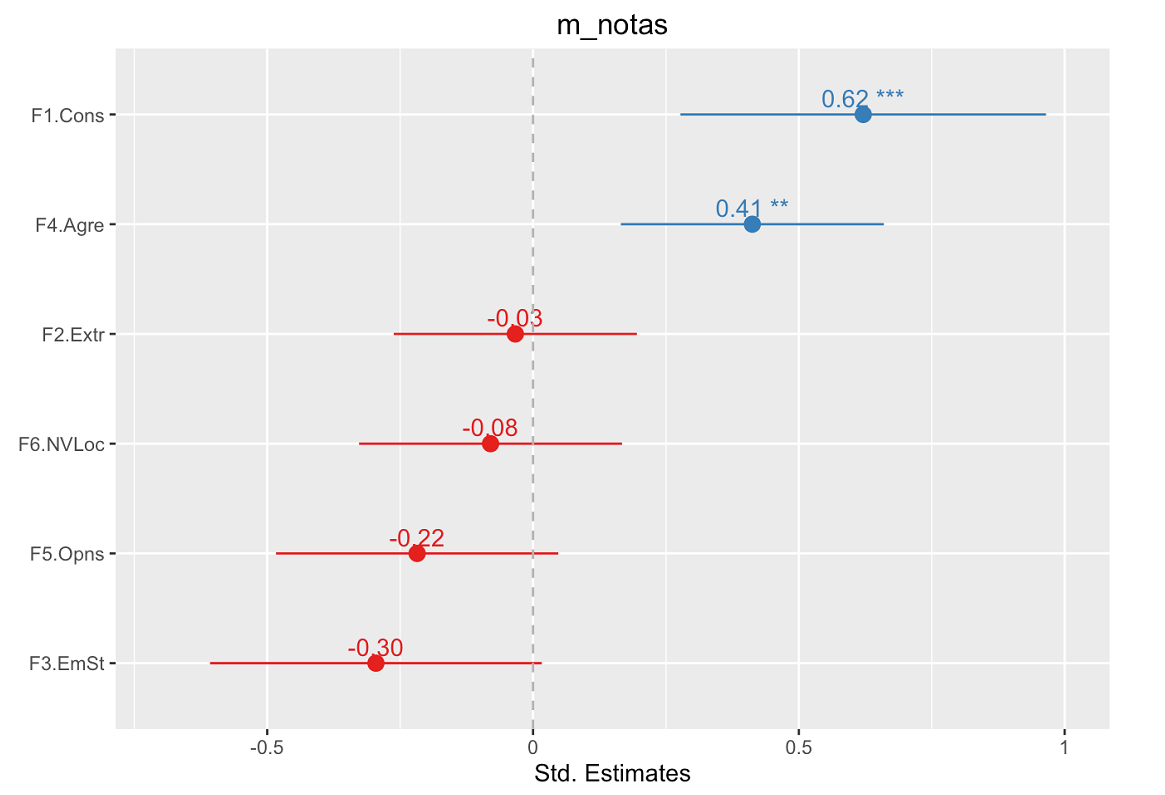
Saída no programa:



Comando:

sjp.lm(fit2, type = "std")

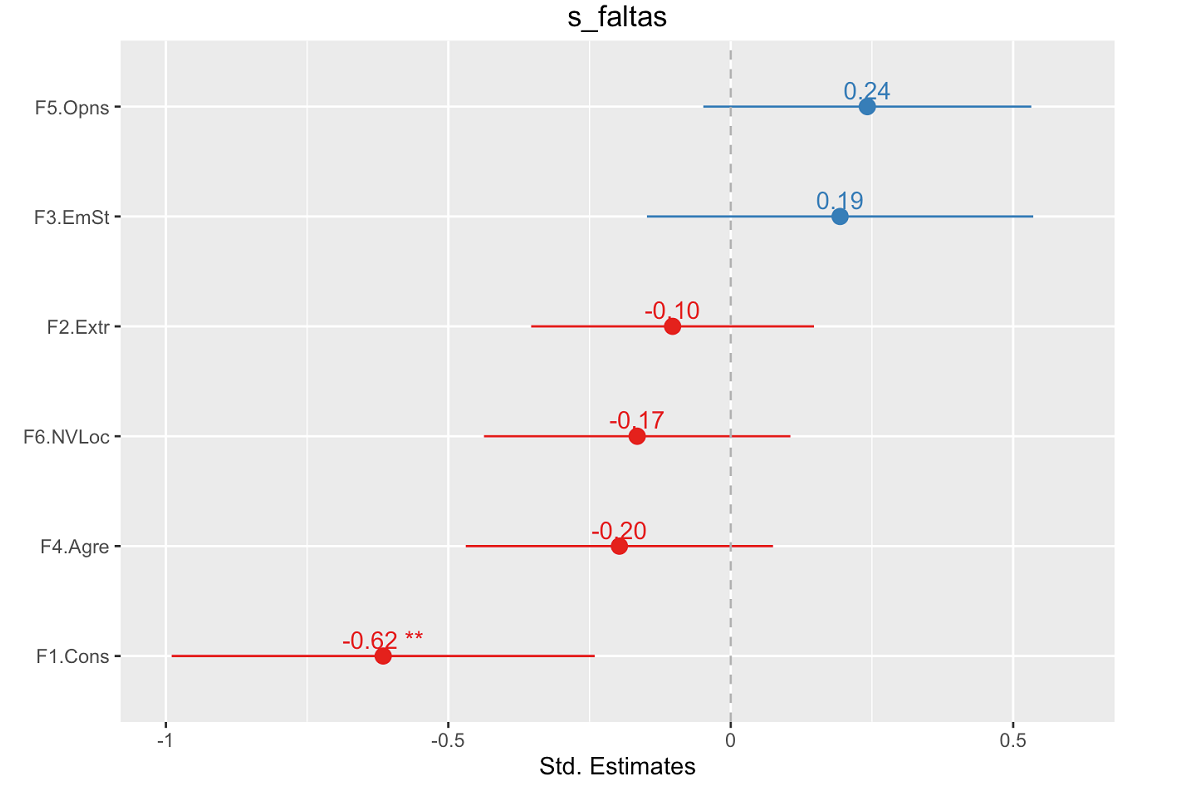
Saída no programa:

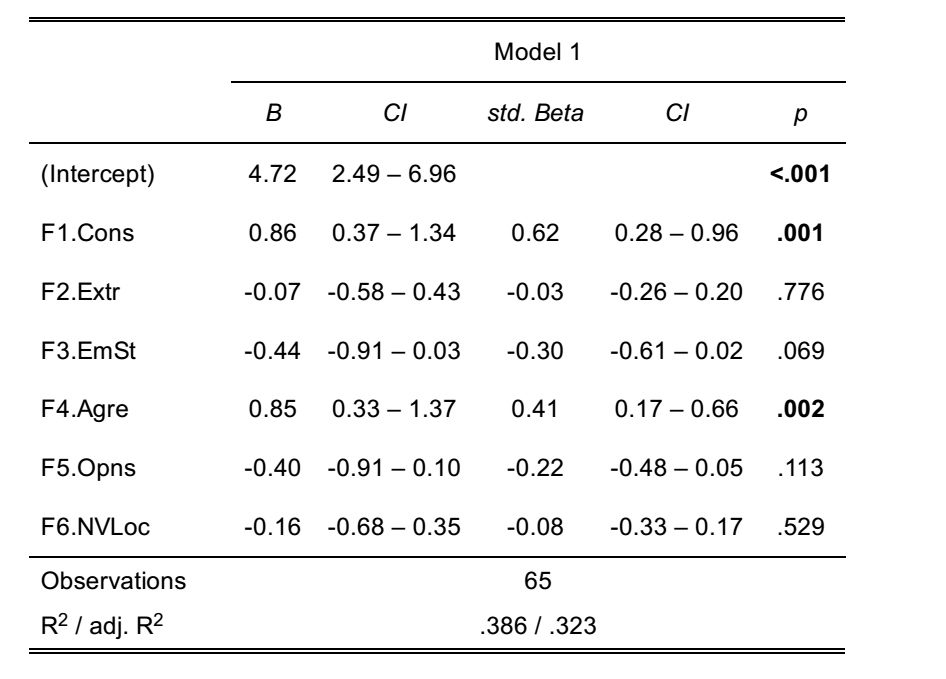


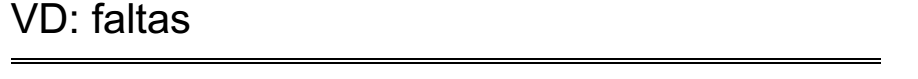
Comando:

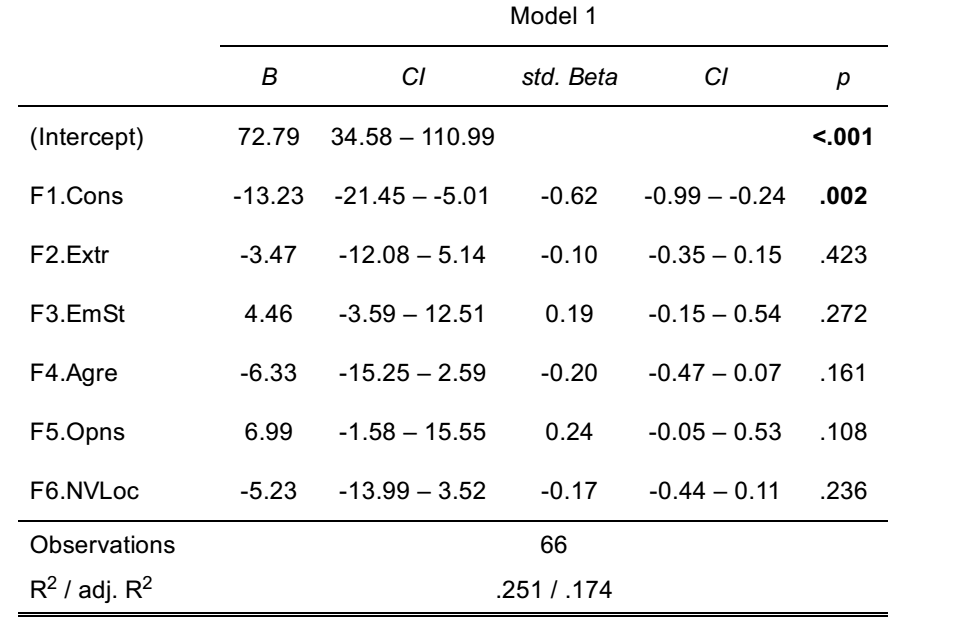
sjp.lm(fit3, type = "std")

Saída no programa:

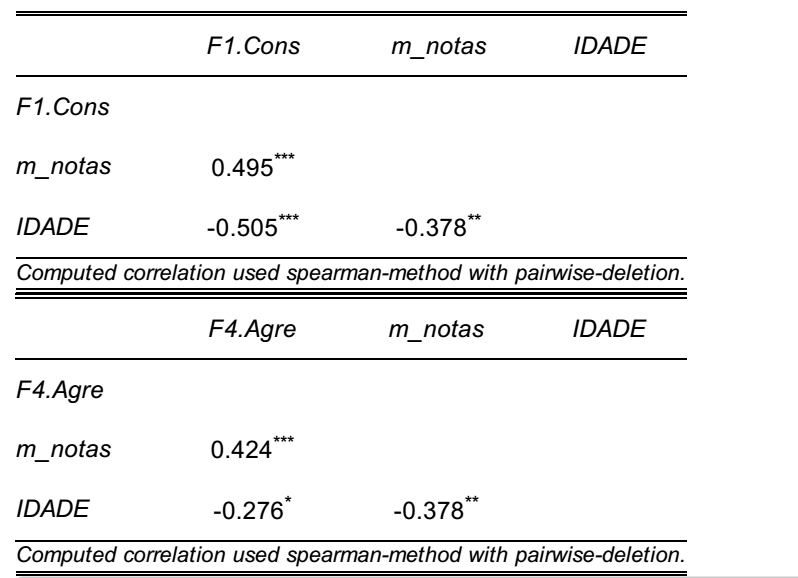








#### **Estudo de padrões desenvolvimentais e auto-gestão: revisitando a relação entre nota, escolaridade e auto-gestão usando regressão múltipla**



Comando:

*# Correlação entre notas e habilidades socioemocionais*

**library**(sjPlot)

sjt.corr(sennav1[ , c("F1.Cons", "m\_notas", "IDADE")], triangle="lower")

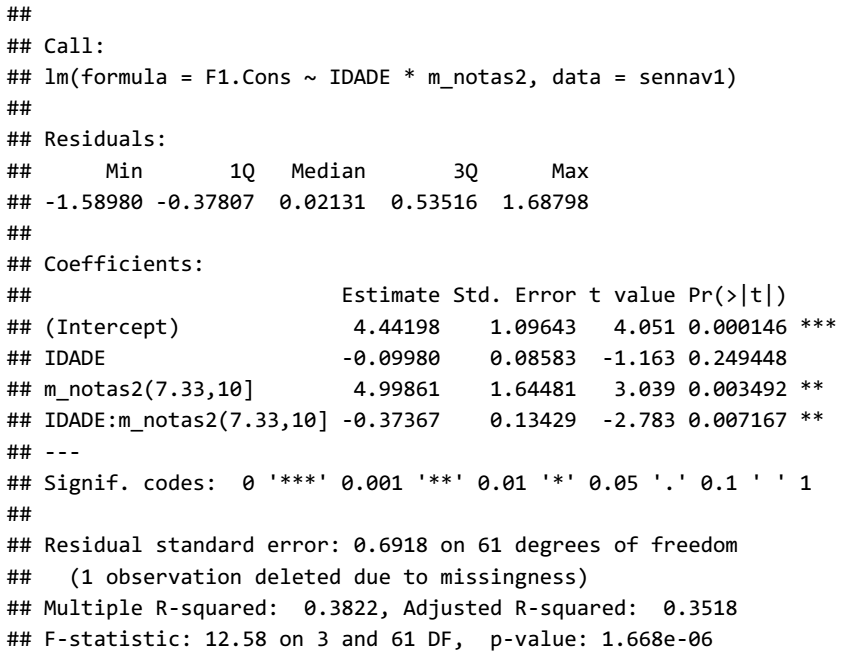
sjt.corr(sennav1[ , c("F4.Agre", "m\_notas", "IDADE")], triangle="lower")

*# Cria escolaridade como uma variável "factor"*

fit4 <- lm(F1.Cons~IDADE\*m\_notas2, data=sennav1)

summary(fit4) *# show results*

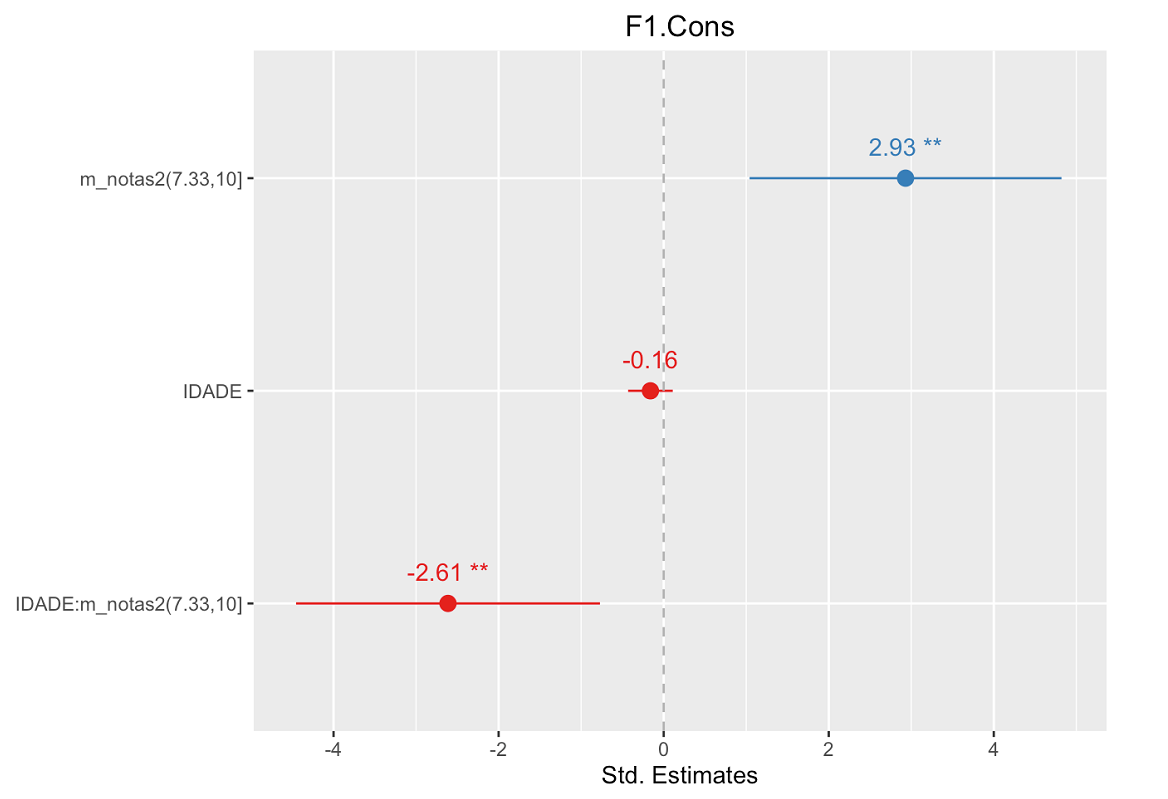
Saída no programa:



Comando:

sjp.lm(fit4, type = "std")

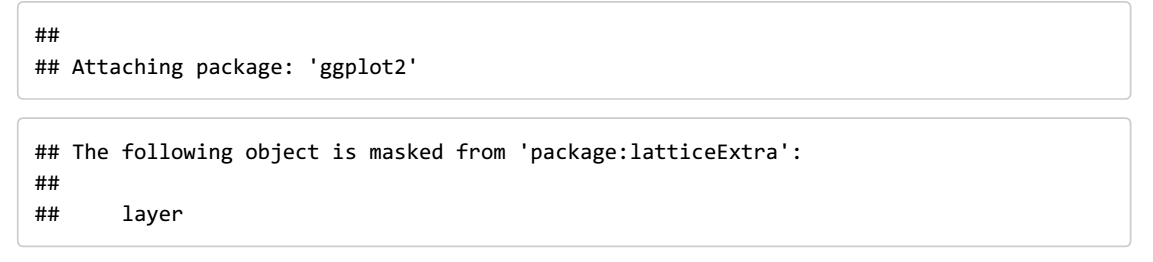
Saída no programa:



Comando:

**library**(ggplot2)

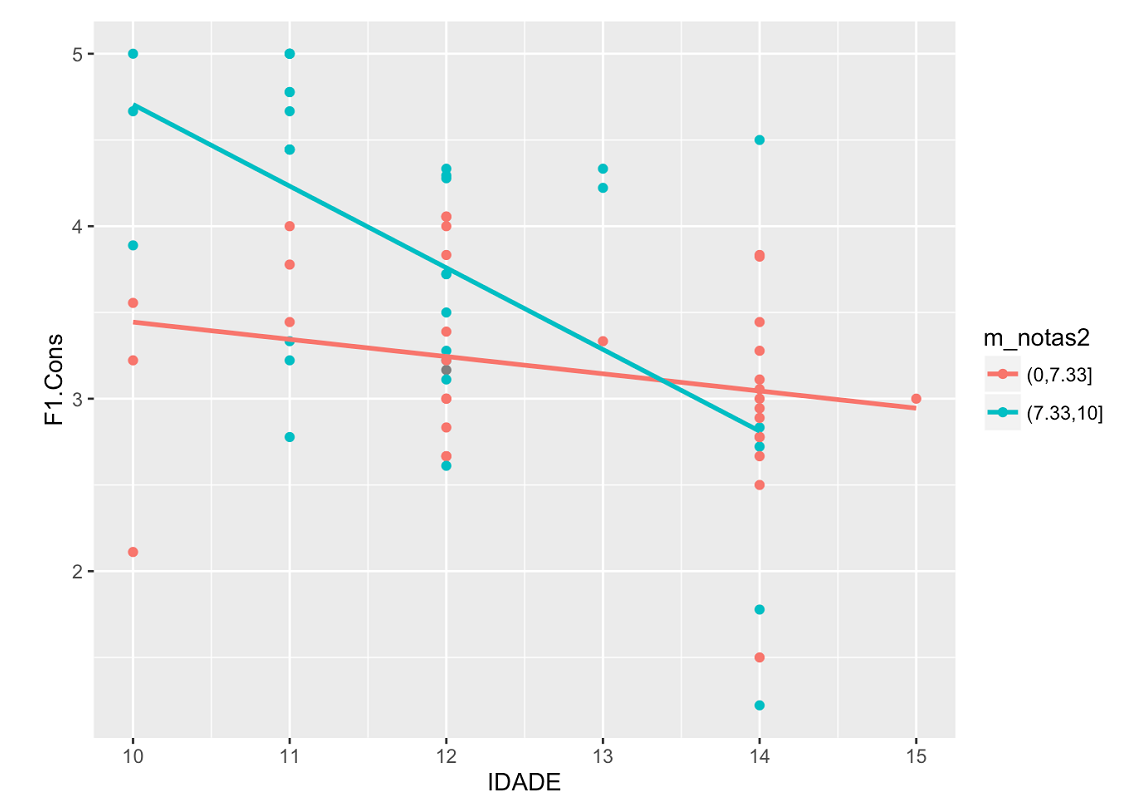
Saída no programa:



Comando:

ggplot(data=sennav1, aes(x=IDADE, y=F1.Cons, color = m\_notas2)) + geom\_point() + geom\_smooth(method="lm", se=FALSE)

Saída no programa:



Comando:

sjt.lm(fit4, showStdBeta = TRUE)

Saída no programa:



#### **Estudo dos coeficientes não padronizados e padronizados na regressão múltipla**

Comando:

*# Separa base so com as variáveis de interesse*

dt <- sennav1[ , c("m\_notas", "F1.Cons", "F2.Extr",

"F3.EmSt", "F4.Agre", "F5.Opns", "F6.NVLoc")]

*# Roda regressão múltipla*

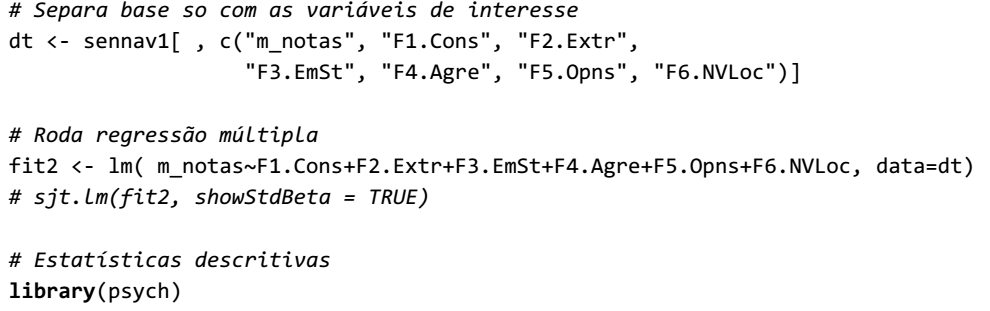
fit2 <- lm( m\_notas~F1.Cons+F2.Extr+F3.EmSt+F4.Agre+F5.Opns+F6.NVLoc, data=dt)

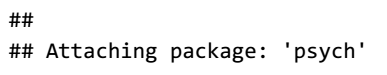
*# sjt.lm(fit2, showStdBeta = TRUE)*

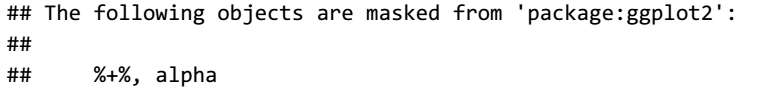
*# Estatísticas descritivas*

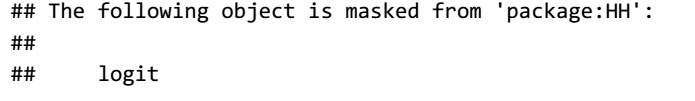
**library**(psych)

Saída no programa:





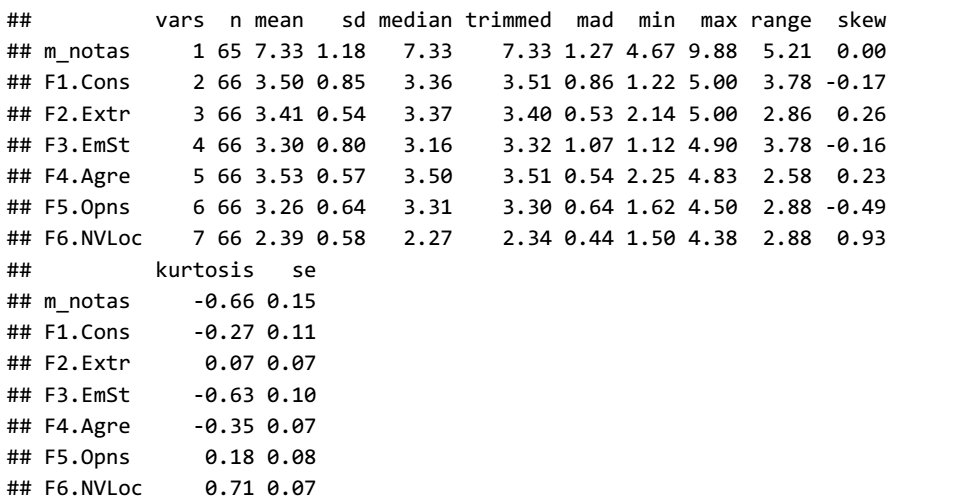




Comando:

describe(dt)

Saída no programa:



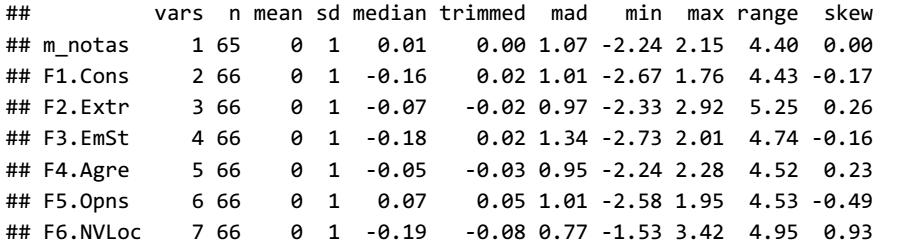
Comando:

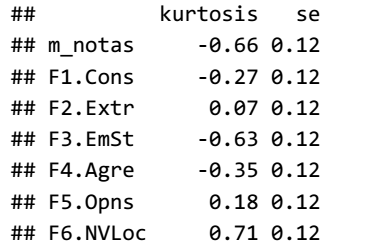
*# Transforma variáveis em z e mostra estatísticas descritivas*

dtz <- as.data.frame(scale(dt))

describe(dtz)

Saída no programa:





Comando:

fit3 <- lm( m\_notas~F1.Cons+F2.Extr+F3.EmSt+F4.Agre+F5.Opns+F6.NVLoc, data=dtz)

sjt.lm(fit3, show.std = TRUE)

*# install.packages("ggExtra")*

**library**(ggplot2)

**library**(ggExtra)

*#*

p <- ggplot(data=dt, aes(x=F1.Cons, y=m\_notas)) +

geom\_point() + geom\_smooth(method="lm") +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas") +

scale\_y\_continuous(breaks=seq(1, 10, 1)) +

scale\_x\_continuous(breaks=seq(1, 5, 0.5)) +

geom\_hline(yintercept=c(7.33, 7.33+1.18, 7.33-1.18),

colour="green", linetype="longdash") +

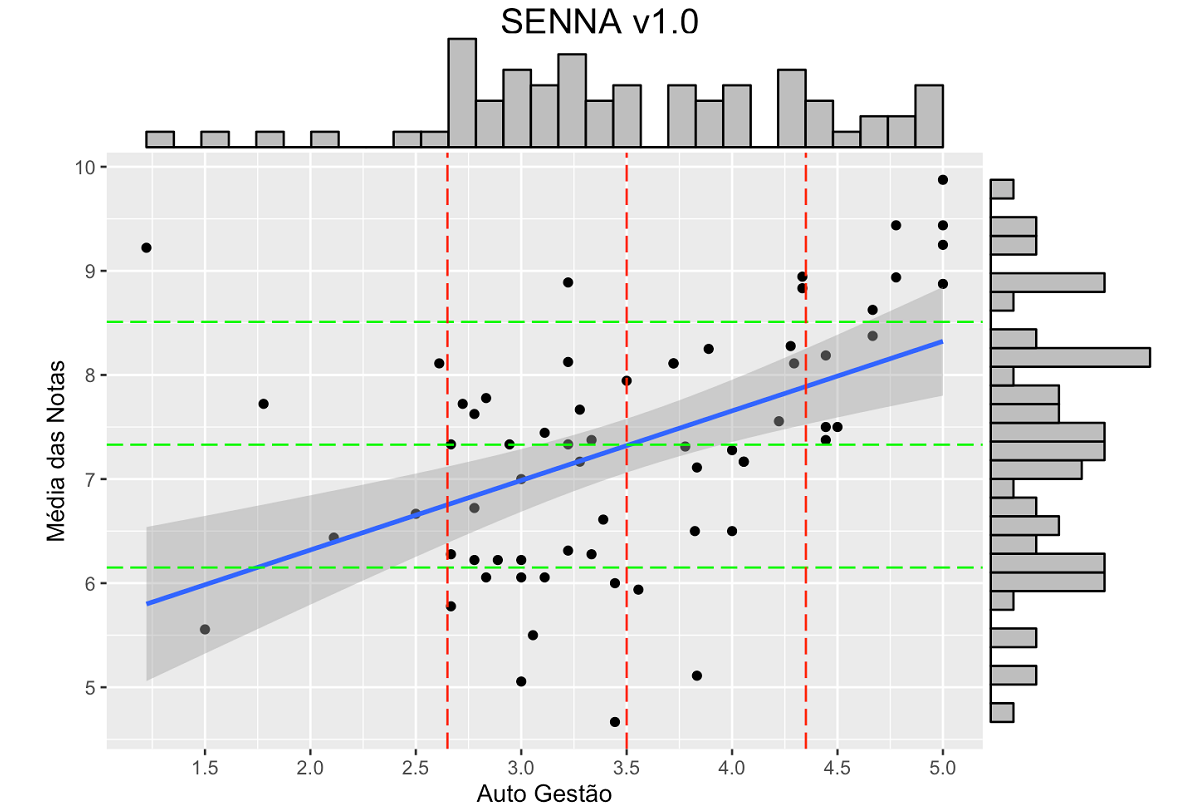
geom\_vline(xintercept=c(3.50, 3.50+0.85, 3.50-0.85),

colour="red", linetype="longdash")

ggMarginal(p, type = "histogram")

Saída no programa:

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Comando:

p2 <- ggplot(data=dtz, aes(x=F1.Cons, y=m\_notas)) +

geom\_point() + geom\_smooth(method="lm") +

labs(title="SENNA v1.0", x="Auto Gestão", y="Média das Notas") +

scale\_y\_continuous(breaks=seq(-3, 3, 0.5)) +

scale\_x\_continuous(breaks=seq(-3, 3, 0.5)) +

geom\_hline(yintercept=c(0, 1, -1),

colour="green", linetype="longdash") +

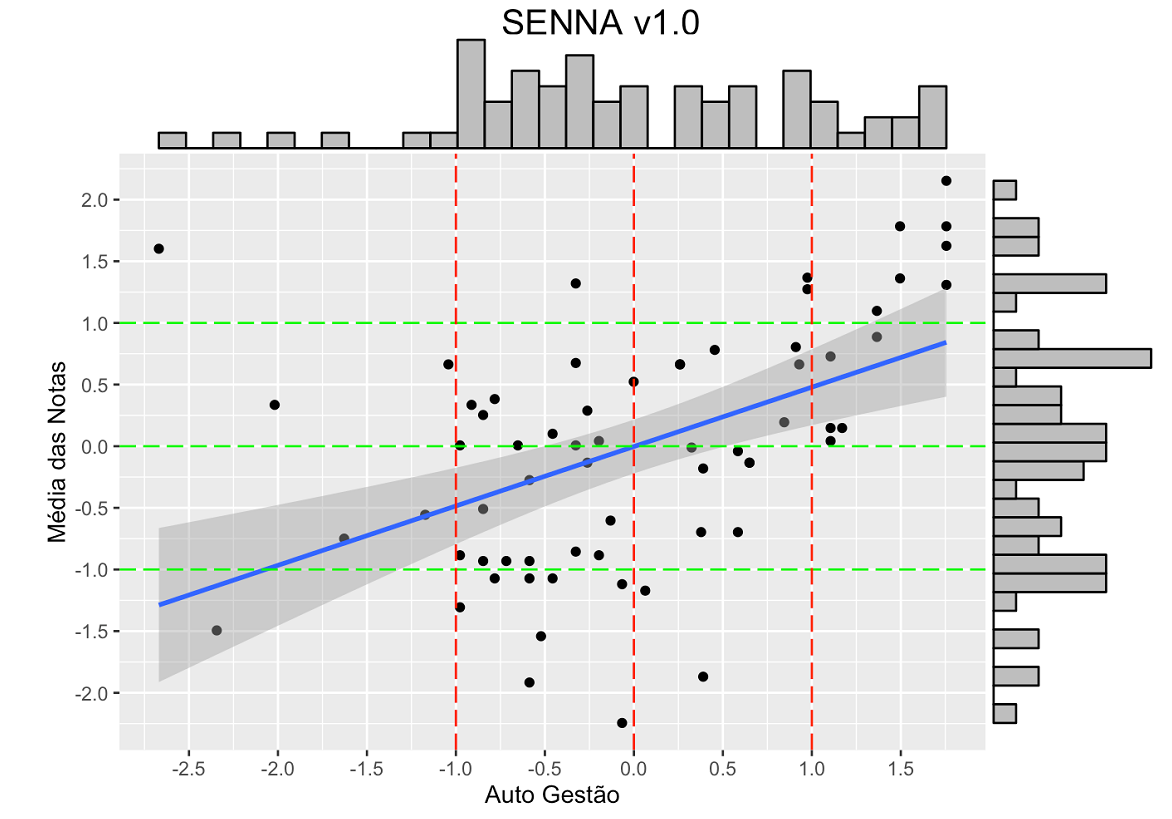
geom\_vline(xintercept=c(0, 1, -1),

colour="red", linetype="longdash")

ggMarginal(p2, type = "histogram")

Saída no programa:





#### **Visualização da regressão múltipla**

Comando:

*# http://www.statmethods.net/graphs/scatterplot.html*

*# 3D Scatterplot with Coloring and Vertical Lines*

*# and Regression Plane*

*# install.packages("scatterplot3d")*

**library**(scatterplot3d)

**attach**(dtz)

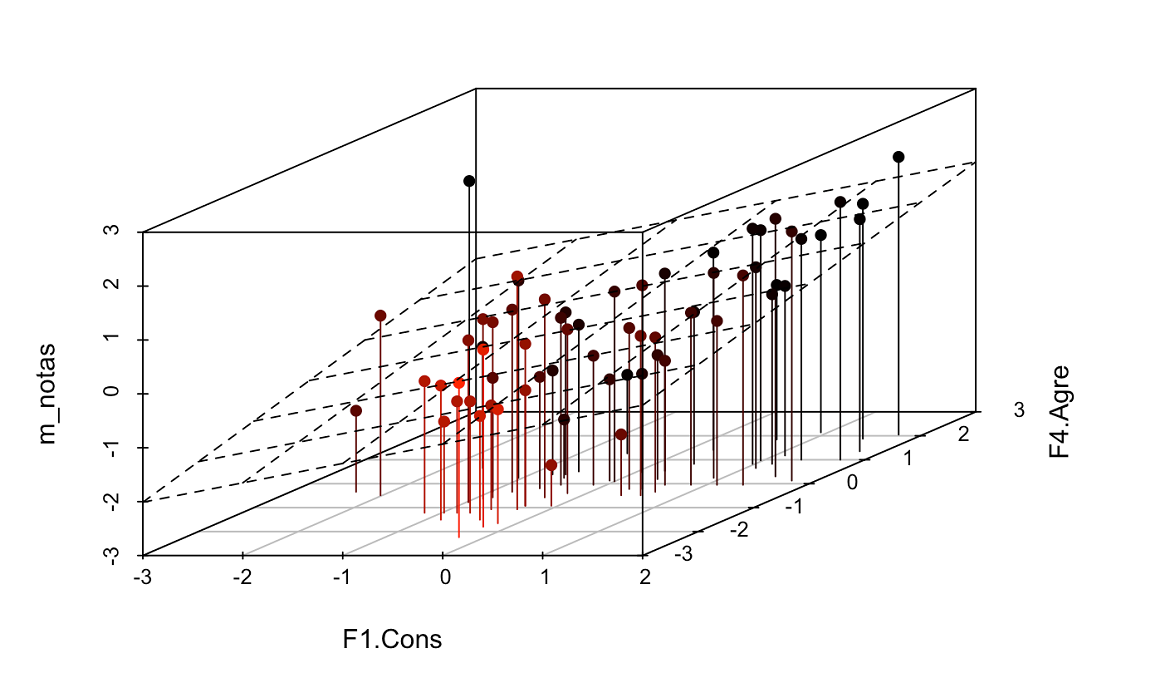
fit <- lm( m\_notas~F1.Cons+F4.Agre)

s3d <-scatterplot3d(F1.Cons, F4.Agre, m\_notas, pch=16,

highlight.3d=TRUE, type="h")

s3d$plane3d(fit)

Saída no programa:



*# install.packages("Rcmdr")*

*#library(Rcmdr)*

*#scatter3d(F1.Cons, m\_notas, F4.Agre)*

*#scatter3d(F1.Cons, m\_notas, F3.EmSt)*

### **Exercício 4**

* Faça uma ANOVA avaliando efeito de IDADE e SEXO em relação a um fator
* Faça uma regressão múltipla verificando o efeito escolaridade do pai e da mãe, sexo e renda nas habilidades que você estudou

1. Por causa dos termos técnicos utilizamos: Walter, O.M.F; Henning, E.; Konrath, A.C; Alves, C.C. & Samohyl, R.W. (2012). Uma visão geral do Rstudio aplicado ao ensino de controle estatístico do processo. Acessado em Maio de 2016 de: http://www.abenge.org.br/CobengeAnteriores/2012/artigos/104106.pdf [↑](#footnote-ref-1)