Anexos

Anexo A: Cálculo de r de Pearson e de r ajustado

Arquivo .R disponível em https://github.com/oushiro/IEL/blob/master/Licao11-demonstracoes.R.

```
###Cálculo do r de Pearson

idade <- c(1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 8, 9, 11, 12, 12)
altura <- c(60, 65, 97, 98, 100, 105, 107, 105, 119, 122, 125, 132, 14
2, 147, 153)

#numero de observacoes
n <- 15
#soma dos valores da variavel x
SX <- sum(idade)
#soma dos valores da variavel y
SY <- sum(altura)
#soma dos valores da variavel x elevados ao quadrado
S.X2 <- sum(idade^2)
#soma dos valores da variavel y elevados ao quadro
S.Y2 <- sum(altura^2)
#soma de x * y
XY <- idade * altura
SXY <- sum(XY)
#r de Pearson
r <- ((n * SXY) - (SX*SY)) / sqrt(((n*S.X2)-(SX^2))*((n*S.Y2)-(SY^2)))
r

#comparar com cor.test:
cor.test(altura, idade)

###Calculo de R^2 ajustado:

#valor de R^2
R2 <- 0.943201
#numero de observacoes
n <- 15
#numero de variaveis independentes / predictors
k <- 1

#formula de R^2 ajustado
1 - (1-R2) * ((n-1)/(n-k-1))
```

Anexo B: Roteiro para análise de variáveis numéricas

Arquivo .R disponível em

https://github.com/oushiro/Introducao_a_Estatistica_para_Linguistas/blob/master/scripts/Licao13-IEL-roteiroAnalise-VRnumerica.R.

```
### Introdução à Estatística para Linguística ###
### L. Oushiro ###
### Roteiro para Análise de Variáveis Numéricas ###

### Preliminares: carregar pacotes e dados; ajustar dados ####
### Carregar pacotes necessários
library(tidyverse)
library(effects)
library(car)
library(lme4)
library(lmerTest)

### Definir diretório de trabalho ####
#setwd()

### Carregar dados ####
pretonicas <- read_csv("Pretonicas.csv",
                       col_types = cols(.default = col_factor(),
                                        VOGAL = col_factor(levels = c(
"i", "e", "a", "o", "u")),
                                        F1 = col_double(),
                                        F2 = col_double(),
                                        F1.NORM = col_double(),
                                        F2.NORM = col_double(),
                                        F1.SIL.SEG = col_double(),
                                        F2.SIL.SEG = col_double(),
                                        F1.SEG.NORM = col_double(),
                                        F2.SEG.NORM = col_double(),
                                        DIST.TONICA = col_double(),
                                        Begin.Time.s = col_double(),
                                        End.Time.s = col_double(),
                                        Duration.ms = col_double(),
                                        IDADE = col_integer(),
                                        IDADE.CHEGADA = col_integer(),
                                        ANOS.SP = col_integer()
                                        )
                       )

### Ajustar dados ####
pretonicas$CONT.PREC <-  fct_collapse(pretonicas$CONT.PREC,
                          dental.alveolar = c("t", "d", "n", "l"),
                          labial = c("p", "b", "m", "f", "v"),
                          palatal.sibilante = c("S", "Z", "L", "s", "z
"),
                          velar = c("k", "g"),
                          vibrante = c("h", "R")
```

```
)

pretonicas$CONT.PREC <- fct_relevel(pretonicas$CONT.PREC, "dental.alve
olar", "labial", "palatal.sibilante", "velar", "vibrante")

pretonicas$CONT.SEG <- fct_collapse(pretonicas$CONT.SEG,
                    dental.alveolar = c("t", "d", "n", "l"),
                    labial = c("p", "b", "m", "f", "v"),
                    palatal.sibilante = c("S", "Z", "L", "N", "s", "z
"),
                    velar = c("k", "g"),
                    vibrante = c("r", "h", "R")
)

pretonicas$CONT.SEG <- fct_relevel(pretonicas$CONT.SEG, "dental.alveol
ar", "labial", "palatal.sibilante", "velar", "vibrante")

### Subconjuntos de dados ####
VOGAL_e <- filter(pretonicas, VOGAL == "e") %>%
  droplevels()

PBSP_e <- filter(pretonicas, VOGAL == "e" & AMOSTRA == "PBSP")

SP2010_e <- filter(pretonicas, VOGAL == "e" & AMOSTRA == "SP2010")

### Checar dados
str(VOGAL_e)
#View(VOGAL_e)
str(PBSP_e)
str(SP2010_e)

### Análises descritivas e univariadas ####
#Cálculo de média, mediana, desvio padrão
pretonicas %>%
  group_by(VOGAL, AMOSTRA) %>%
  summarize(media_F1 = mean(F1.NORM),
            mediana_F1 = median(F1.NORM),
            sd_F1 = sd(F1.NORM))

# Espaços vocálicos de PBSP e SP2010
ggplot(medias, aes(x = media_F2, y = media_F1, color = AMOSTRA, label
= VOGAL)) +
  geom_line() +
  geom_label() +
  scale_x_reverse() +
  scale_y_reverse() +
  ggtitle("Valores médios de F1 e F2 normalizados nas amostras PBSP e
SP2010") +
  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()

### F1.NORM ~ AMOSTRA (vogal /e/)
#Teste-t
```

```
# Checar normalidade da distribuição com Teste de Shapiro
shapiro.test(PBSP_e$F1.NORM)
shapiro.test(SP2010_e$F1.NORM)
# t.test(F1.NORM ~ AMOSTRA, data = VOGAL_e)
wilcox.test(F1.NORM ~ AMOSTRA, data = VOGAL_e, conf.int = T)

### Outras vogais?... Outras variáveis previsoras?...

# Gráfico de dispersão das medições de F1 e F2 normalizados em PBSP e
SP2010
ggplot(pretonicas, aes(x = F2.NORM, y = F1.NORM, color = VOGAL)) +
  geom_point() +
  scale_x_reverse() +
  scale_y_reverse() +
  facet_grid(. ~ AMOSTRA) +
  ggtitle("Dispersão das medidas de F1 e F2 normalizados nas amostras
PBSP e SP2010") +
  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()

# Boxplots das medidas de F1.NORM por VOGAL e por AMOSTRA
ggplot(pretonicas, aes(x = AMOSTRA, y = F1.NORM, color = VOGAL)) +
  geom_boxplot(notch = TRUE) +
  scale_y_reverse() +
  labs(x = "Amostra", y = "F1 normalizado") +
  facet_grid(. ~ VOGAL) +
  theme_bw()

# Histograma das medições de F1.NORM das cinco vogais dos dados de PBS
P e SP2010
ggplot(pretonicas, aes(x = F1.NORM, fill = AMOSTRA)) +
  geom_histogram(binwidth = 10, position = "identity", alpha = 0.4) +
  labs(x = "F1 normalizado", y = "Frequência") +
  facet_grid(VOGAL ~ .) +
  theme_bw()

# Scatterplot de F1.NORM por F1.SEG.NORM
ggplot(VOGAL_e, aes(x = F1.SEG.NORM, y = F1.NORM)) +
  geom_point() +
  scale_y_reverse() +
  facet_grid(. ~ AMOSTRA) +
  geom_smooth(method = "lm", se = TRUE, color = "lightgrey")

#Teste de correlação de Pearson
# Checar normalidade das distribuições com Teste de Shapiro
shapiro.test(VOGAL_e$F1.NORM)
shapiro.test(VOGAL_e$F1.SEG.NORM)

cor.test(VOGAL_e$F1.NORM, VOGAL_e$F1.SEG.NORM, method = "spearman") #
para distribuição não normal - teste não paramétrico

#Modelo de regressão linear
mod <- lm(F1.NORM ~ F1.SEG.NORM, data = VOGAL_e)
```

```
summary(mod)
plot(effect("F1.SEG.NORM", mod), grid = T, ylim = c(460, 410))

### Análises multivariadas ####

### Modelo de regressão linear ####
mod <- lm(F1.NORM ~ AMOSTRA + SEXO + F1.SEG.NORM + CONT.PREC + CONT.SE
G, data = VOGAL_e2)
summary(mod)

### Função step() ####
m0 <- lm(F1.NORM ~ 1, data = VOGAL_e2)
m.fw <- step(m0, direction = "forward", scope = ~ AMOSTRA + SEXO + F1.
SEG.NORM + CONT.PREC + CONT.SEG)
m.fw

m.bw <- step(mod, direction = "backward")
m.bw

m.both <- step(m0, scope = ~ AMOSTRA + SEXO + F1.SEG.NORM + CONT.PREC
+ CONT.SEG)
m.both

### Função drop1() ####
drop1(mod, test = "F")

### Novo modelo linear sem variável SEXO
modelo <- lm(F1.NORM ~ F1.SEG.NORM + AMOSTRA + CONT.SEG + CONT.PREC, d
ata = VOGAL_e2)
summary(modelo)

### Checagem de pressupostos ####

### (a) relação entre variável resposta e variável previsora numérica
é linear?
# Aplicar função crPlot() (depende do pacote car)
crPlot(modelo, var = "F1.SEG.NORM") # valores atípicos F1.SEG.NORM > 5
00Hz

VOGAL_e3 <- filter(VOGAL_e2, F1.SEG.NORM < 500)

# Novo modelo linear
modelo2 <- lm(F1.NORM ~ AMOSTRA + F1.SEG.NORM + CONT.PREC + CONT.SEG,
data = VOGAL_e3)
summary(modelo2)
crPlot(modelo2, var = "F1.SEG.NORM")

### (b) Há multicolinearidade? (função vif() depende do pacote car)
vif(modelo2)

### (c) Resíduos têm distribuição normal?
shapiro.test(modelo2$residuals)
```

```
### (d) Observações são independentes? -- em dados linguísticos, quase
nunca são! --> MODELOS DE EFEITOS MISTOS

### Criar modelo linear de efeitos mistos ###
# função lmer() depende dos pacotes lme4 e lmerTest
mod1.lmer <- lmer(F1.NORM ~ AMOSTRA + F1.SEG.NORM + CONT.PREC + CONT.S
EG + (1|INFORMANTE) + (1|PALAVRA), data = VOGAL_e3)
summary(mod1.lmer)

# Função step backward
m.bw.lmer <- step(mod1.lmer, direction = "backward")
m.bw.lmer

### ~  Plotar estimativas do modelo: plot_model() ~ ####
#https://cran.r-project.org/web/packages/sjPlot/vignettes/plot_model_e
stimates.html (requer pacote sjPlot); somente para efeitos fixos
library(sjPlot)

plot_model(mod, transform = NULL, show.values = T, value.offset = .3)
```

Anexo C: Cálculo do valor de significância para modelo

Arquivo .R disponível em

https://github.com/oushiro/Introducao_a_Estatistica_para_Linguistas/blob/master/scripts/Licao14-IEL-calculoSignificanciaModelo.R.

```
## Cálculo do valor de significância para modelo logístico como um todo

# visualizar dados do modelo
modelo$null.deviance
modelo$deviance
modelo$df.null
modelo$df.residual

# Função pchisq faz teste de qui-quadrado e cálculo de valor-p a partir dos parâmetros acima
pchisq(modelo$null.deviance - modelo$deviance, modelo$df.null - modelo$df.residual, lower.tail=F)
```

Anexo D: Roteiro para análise de variáveis nominais binárias

Arquivo .R disponível em

https://github.com/oushiro/Introducao_a_Estatistica_para_Linguistas/blob/master/scripts/Licao15-IEL-roteiroAnalise-VRnominal.R.

```
### Introdução à Estatística para Linguística ###
### L. Oushiro ###
### Roteiro para Análise de Variáveis Nominais Binárias ###

### Preliminares: carregar pacotes e dados; ajustar dados ####
### Carregar pacotes necessários ####
library(rms)
library(effects)
library(car)
library(lme4)
library(lmerTest)

### Definir diretório de trabalho ####
# setwd()

### Carregar dados ####
dados <- read_csv("DadosRT.csv",
                  col_types = cols(.default = col_factor(),
                                   VD = col_factor(levels = c("tepe",
"retroflexo")),
                                   FAIXA.ETARIA = col_factor(levels =
c("1a", "2a", "3a")),
                                   ESCOLARIDADE = col_factor(levels =
c("fundamental", "medio", "superior")),
                                   REGIAO = col_factor(levels = c("cen
tral", "periferica")),
                                   CONT.FON.PREC = col_factor(levels =
c("i", "e", "3", "a", "0", "o", "u")),
                                   TONICIDADE = col_factor(levels = c(
"atona", "tonica")),
                                   POSICAO.R = col_factor(levels = c("
final", "medial")),
                                   CLASSE.MORFOLOGICA = col_factor(lev
els = c("adjetivo", "adverbio", "conj.prep", "morf.inf", "substantivo"
, "verbo")),
                                   IDADE = col_integer(),
                                   INDICE.SOCIO = col_double(),
                                   FREQUENCIA = col_double()
                  )
)

### Ajustar dados ####
dados$CONT.FON.SEG <- fct_collapse(dados$CONT.FON.SEG,
                                   pausa = "#",
                                   coronal = c("t", "d", "s", "z", "x"
, "j", "ts", "dz", "l", "n"),
```

```
                                    labial = c("p", "b", "f", "v", "m")
,
                                    dorsal = c("k", "g", "h")
)

dados$CONT.FON.SEG <- fct_relevel(dados$CONT.FON.SEG, "pausa", "corona
l", "dorsal", "labial")

### Checar dados ####
str(dados)
View(dados)

### Análises descritivas e univariadas ####
# VD ####
dados %>%
  count(VD) %>%
  mutate(prop = prop.table(n))

# VD ~ SEXO.GENERO ####
tab.prop.SEXO.GENERO <- dados %>%
  count(SEXO.GENERO, VD) %>%
  group_by(SEXO.GENERO) %>%
  mutate(prop = prop.table(n)) %>%
  print()

ggplot(tab.prop.SEXO.GENERO, aes(x = SEXO.GENERO, y = prop * 100, fill
= VD)) +
  geom_bar(stat = "identity", color = "black") +
  ggtitle("Proporção das variantes de /r/ por Sexo/Gênero do falante")
+
  labs(x = "Sexo", y = "Proporção", fill = "Variantes de /r/") +
  scale_x_discrete(labels = c("feminino", "masculino")) +
  scale_fill_brewer(palette = "Purples", labels = c("tepe", "retroflex
o")) +
  theme_bw()

tab.SEXO.GENERO <- with(dados, table(SEXO.GENERO, VD)); tab.SEXO.GENER
O
prop.SEXO.GENERO <- with(dados, prop.table(tab.SEXO.GENERO) * 100); pr
op.SEXO.GENERO

chisq.test(tab.SEXO.GENERO)
mod <- glm(VD ~ SEXO.GENERO, data = dados, family = binomial)
summary(mod)
plot(allEffects(mod), type = "response")

# VD ~ FAIXA.ETARIA ####
tab.prop.FAIXA.ETARIA <- dados %>%
  count(FAIXA.ETARIA, VD) %>%
  group_by(FAIXA.ETARIA) %>%
  mutate(prop = prop.table(n)) %>%
  print()
```

```
ggplot(tab.prop.FAIXA.ETARIA, aes(x = FAIXA.ETARIA, y = prop * 100, fi
ll = VD)) +
  geom_bar(stat = "identity", color = "black") +
  ggtitle("Proporção das variantes de /r/ por Faixa Etária do falante"
) +
  labs(x = "Faixa Etária", y = "Proporção", fill = "Variantes de /r/")
+
  scale_x_discrete(labels = c("20-34", "35-59", "60+")) +
  scale_fill_brewer(palette = "Purples", labels = c("tepe", "retroflex
o")) +
  theme_bw()

tab.prop.FAIXA.ETARIA %>%
  filter(VD == "retroflexo") %>%
  ggplot(., aes(x = FAIXA.ETARIA, y = prop * 100, group = VD)) +
  geom_line(linetype = "dotted", size = 1, color = "blue") +
  geom_point(shape = 18, size = 3, fill = "black") +
  ggtitle("Proporção de retroflexo por Faixa Etária do falante") +
  labs(x = "Faixa Etária", y = "Proporção", fill = "Variantes de /r/")
+
  scale_x_discrete(labels = c("1a", "2a", "3a")) +
  ylim(0, 50) +
  theme_bw()

tab.FAIXA.ETARIA <- with(dados, table(FAIXA.ETARIA, VD)); tab.FAIXA.ET
ARIA
prop.FAIXA.ETARIA <- with(dados, prop.table(tab.FAIXA.ETARIA) * 100);
prop.FAIXA.ETARIA

chisq.test(tab.FAIXA.ETARIA)
chisq.test(tab.FAIXA.ETARIA[c(2, 3), ]) # 2a vs 3a
mod <- glm(VD ~ FAIXA.ETARIA, data = dados, family = binomial)
summary(mod)
plot(allEffects(mod), type = "response")

# VD ~ INDICE.SOCIO ####
mod <- glm(VD ~ INDICE.SOCIO, data = dados, family = binomial)
summary(mod)
plot(allEffects(mod), type = "response")

# Outras variáveis?... ESCOLARIDADE, REGIAO, ORIGEM.PAIS, CONT.FON.PRE
C...

### Análises multivariadas ####
# Checar ortogonalidade entre variáveis previsoras
# CONT.FON.SEG e POSICAO.R
with(dados, table(CONT.FON.SEG, POSICAO.R))

# CLASSE.MORFOLOGICA e POSICAO.R
with(dados, table(CLASSE.MORFOLOGICA, POSICAO.R))

# CLASSE.MORFOLOGICA e TONICIDADE
with(dados, table(CLASSE.MORFOLOGICA, TONICIDADE))
```

```
# Modelo com TONICIDADE, POSICAO.R, CLASSE.MORFOLOGIA e CONT.FON.SEG p
ara verificar multicolinearidade
mod <- glm(VD ~
             TONICIDADE +
             POSICAO.R +
             CLASSE.MORFOLOGICA +
             CONT.FON.SEG,
           data = dados, family = binomial)

summary(mod)

# Função vif() para avaliar multicolinearidade (requer pacote car)
car::vif(mod)

### Modelo de regressão logística ####
modelo <- glm(VD ~
                SEXO.GENERO +
                FAIXA.ETARIA * REGIAO +
                INDICE.SOCIO +
                CONT.FON.PREC +
                CONT.FON.SEG +
                TONICIDADE +
                POSICAO.R +
                CLASSE.MORFOLOGICA,
              data = dados, family = binomial)

summary(modelo)
lrm(VD ~
      SEXO.GENERO +
      FAIXA.ETARIA * REGIAO +
      INDICE.SOCIO +
      CONT.FON.PREC +
      CONT.FON.SEG +
      TONICIDADE +
      POSICAO.R +
      CLASSE.MORFOLOGICA,
    data = dados)

### Função step() ####
m0 <- glm(VD ~ 1, data = dados, family = binomial)
m.fw <- step(m0,
             direction = "forward",
             scope = ~
               SEXO.GENERO +
               FAIXA.ETARIA * REGIAO +
               INDICE.SOCIO +
               CONT.FON.PREC +
               CONT.FON.SEG +
               TONICIDADE +
               POSICAO.R +
               CLASSE.MORFOLOGICA)
m.fw
```

```
m.bw <- step(modelo, direction = “backward”)
m.bw

m.both <- step(m0, scope = ~
                  SEXO.GENERO +
                  FAIXA.ETARIA * REGIAO +
                  INDICE.SOCIO +
                  CONT.FON.PREC +
                  CONT.FON.SEG +
                  TONICIDADE +
                  POSICAO.R +
                  CLASSE.MORFOLOGICA)
m.both

### Função drop1() ####
drop1(modelo, test = “LR”)

### Testar overfitting
mod.lrm <- lrm (VD ~
                   SEXO.GENERO +
                   FAIXA.ETARIA * REGIAO +
                   INDICE.SOCIO +
                   CONT.FON.PREC +
                   CONT.FON.SEG +
                   TONICIDADE +
                   POSICAO.R +
                   CLASSE.MORFOLOGICA,
                 data = dados, x = T, y = T)

### Função validate() - requer pacote rms
validate(mod.lrm, B = 200, bw = T)

### Checagem de pressupostos ####

### (a) A relação entre o logit e as variáveis previsoras numéricas é
linear?
# Fazer modelo sem interação para aplicar crPlot
modelo2 <- glm(VD ~
                  SEXO.GENERO +
                  FAIXA.ETARIA +
                  REGIAO +
                  INDICE.SOCIO +
                  CONT.FON.PREC +
                  CONT.FON.SEG +
                  TONICIDADE +
                  POSICAO.R +
                  CLASSE.MORFOLOGICA,
                data = dados, family = binomial)

# Aplicar crPlot() ao modelo (requer pacote car)
crPlot(modelo2, variable = “INDICE.SOCIO”)
```

```
### (b) Há multicolinearidade?
car::vif(modelo)

### (c) Observações são independentes? -- em dados linguísticos, quase
nunca são! --> MODELOS DE EFEITOS MISTOS

### Criar modelo linear de efeitos mistos ###
# função glmer() depende dos pacotes lme4 e lmerTest
mod.glmer <- glmer(VD ~
                     SEXO.GENERO +
                     FAIXA.ETARIA * REGIAO +
                     INDICE.SOCIO +
                     CONT.FON.PREC +
                     CONT.FON.SEG +
                     TONICIDADE +
                     POSICAO.R +
                     CLASSE.MORFOLOGICA +
                     (1|INFORMANTE) +
                     (1|ITEM.LEXICAL),
                   data = dados, family = binomial)

# Aplicar summary() a mod.glmer
summary(mod.glmer)

# Visualizar resultados numéricos em gráfico de efeitos (requer pacote
effects)
plot(allEffects(mod.glmer), type = "response")

# Gráficos de efeitos com argumento ask = T
plot(allEffects(mod.glmer), type = "response", ask = T)
```