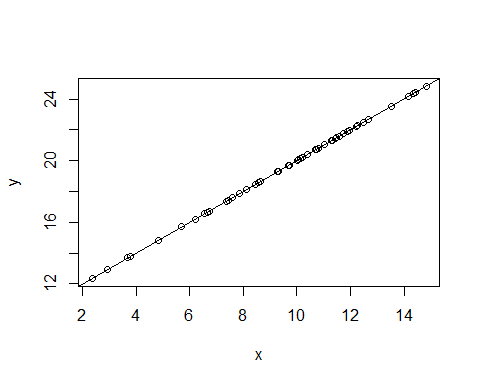
R lecture 2023 01 17 (multilevel models)

Adam Tallman

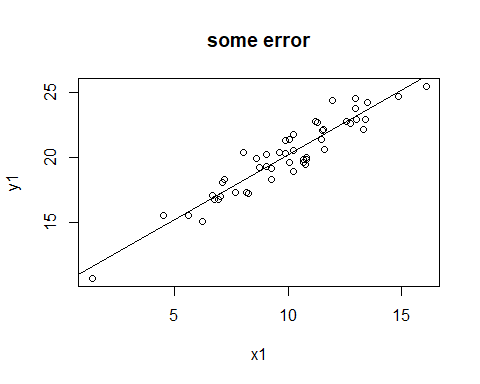
2023-01-16

## Linear model (perfect prediction)

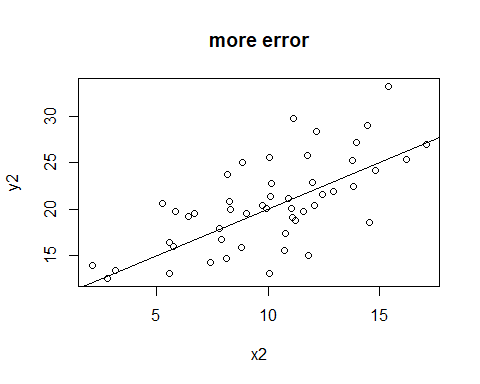
x <- rnorm(50, m =10, sd = 3)  
a = 10  
b = 1  
y <- a+b\*x   
plot(y~x)  
abline(coef(lm(y~x)))



x1 <- rnorm(50, m =10, sd = 3)  
a1 = 10  
b1 =1  
e1 <- rnorm(n=50, m=0, sd=1) #Add an error term  
y1 <- a1+b1\*x1 +e1  
data1 <- data.frame(x1,y1)  
plot(y1~x1)  
abline(coef(lm(y1~x1)))  
title("some error")



x2 <- rnorm(50, m =10, sd = 3)  
a2 = 10  
b2 =1  
e2 <- rnorm(n=50, m=0, sd=4)  
y2 <- a2+b2\*x2 +e2  
data2 <- data.frame(x2,y2)  
data2$group = "large.residuals"  
plot(y2~x2)  
abline(coef(lm(y~x)))  
title("more error")



ldt <- read.csv("/Users/Adan Tallman/Desktop/levshina.ldt.csv", header=TRUE)  
  
head(ldt)

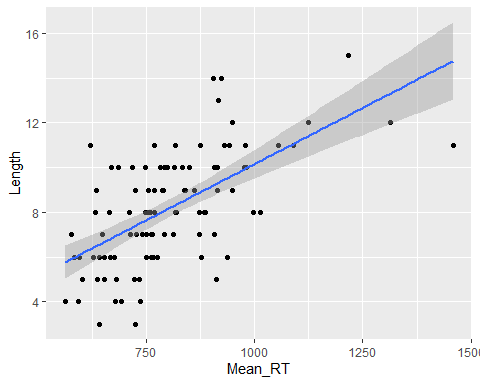
## X Length Freq Mean\_RT  
## 1 marveled 8 131 819.19  
## 2 persuaders 10 82 977.63  
## 3 midmost 7 0 908.22  
## 4 crutch 6 592 766.30  
## 5 resuspension 12 2 1125.42  
## 6 efflorescent 12 9 948.33

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.5.0   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

ggplot(ldt, aes(Mean\_RT, Length))+  
 geom\_point()+  
 geom\_smooth(method="lm")

## `geom\_smooth()` using formula = 'y ~ x'



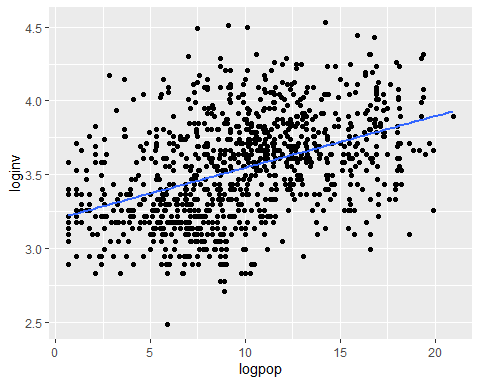
## Phoneme inventory

df <- read.csv("/Users/Adan Tallman/Desktop/inventories.clean.csv", header=TRUE)

It looks like there is a positive correlation between log population and log phoneme inventory. As the population of speakers increases the number of phonemes of their respective language increases.

plot.pooleddata <- ggplot(df, aes(x=logpop, y=loginv))+  
 geom\_point()+  
 geom\_smooth(method="lm",se=FALSE)  
plot.pooleddata

## `geom\_smooth()` using formula = 'y ~ x'



Let’s create a model with log population versus log phoneme inventory.

model.ols1<-lm(loginv~logpop, data=df) # ols = ordinary least squares regression  
summary(model.ols1)

##   
## Call:  
## lm(formula = loginv ~ logpop, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.91777 -0.21683 -0.00495 0.21412 1.02983   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.197035 0.024627 129.82 <2e-16 \*\*\*  
## logpop 0.034921 0.002296 15.21 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3159 on 1001 degrees of freedom  
## Multiple R-squared: 0.1877, Adjusted R-squared: 0.1869   
## F-statistic: 231.3 on 1 and 1001 DF, p-value: < 2.2e-16

## Subsetting by language

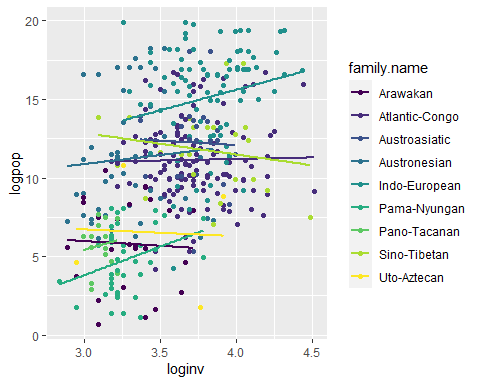
df.lgsubset <- subset(df, family.name == "Indo-European"|  
 family.name =="Uto-Aztecan"|  
 family.name == "Atlantic-Congo"|  
 family.name == "Sino-Tibetan"|  
 family.name == "Otomanguean"|  
 family.name == "Austronesian"|  
 family.name == "Pama-Nyungan"|  
 family.name == "Austroasiatic"|  
 family.name =="Arawakan"|  
 family.name == "Pano-Tacanan")

library(viridis)

## Loading required package: viridisLite

df.lgsubset %>%  
 ggplot(aes(x=loginv,  
 y=logpop,  
 color=family.name))+  
 geom\_point()+  
 geom\_smooth(method="lm",se = FALSE)+  
 scale\_colour\_viridis\_d()

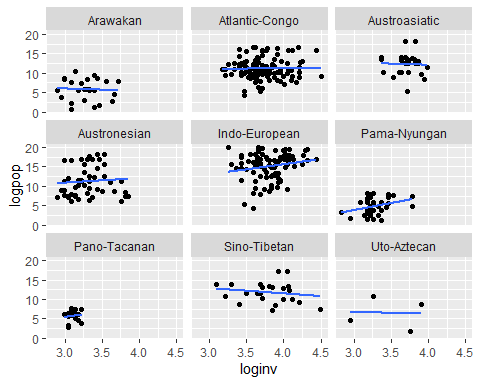
## `geom\_smooth()` using formula = 'y ~ x'



Another way we can see the variation in groups is by looking at regression models for each group.

plot <- ggplot(df.lgsubset, aes(x=loginv, y=logpop, group=family.name))+  
 geom\_point()+  
 geom\_smooth(method="glm",se = FALSE)  
plot + facet\_wrap(~ family.name, ncol=3)

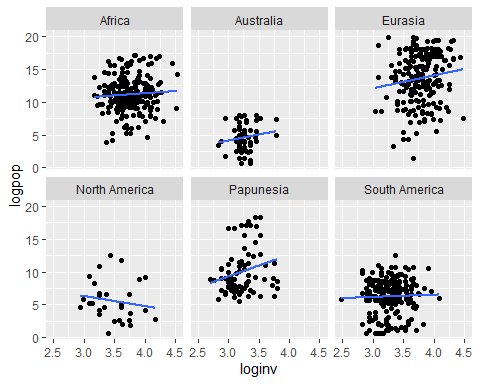
## `geom\_smooth()` using formula = 'y ~ x'



We can also look at differences between areas.

df.areasubset <- subset(df, area !="")  
plot <- ggplot(df.areasubset, aes(x=loginv, y=logpop, group=area))+  
 geom\_point()+  
 geom\_smooth(method="glm",se = FALSE)  
plot + facet\_wrap(~ area, ncol=3)

## `geom\_smooth()` using formula = 'y ~ x'



library(lme4)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

library(lmerTest)

##   
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':  
##   
## lmer

## The following object is masked from 'package:stats':  
##   
## step

## Multilevel model

One disadvantage of multilevel models is that their complexity makes them hard to interpret.

Its typical to AICs in order to assess multilevel models. These are multilevel models without the predictor variable.

mod.lmer.null1 <- lmer(loginv~(1|family.name)+ (1|area), data=df)  
mod.lmer.null2 <- lmer(loginv~(1|family.name), data=df)  
mod.lmer.null3 <- lmer(loginv~(1|area), data=df)

anova(mod.lmer.null1,  
 mod.lmer.null2,  
 mod.lmer.null3)

## refitting model(s) with ML (instead of REML)

## Data: df  
## Models:  
## mod.lmer.null2: loginv ~ (1 | family.name)  
## mod.lmer.null3: loginv ~ (1 | area)  
## mod.lmer.null1: loginv ~ (1 | family.name) + (1 | area)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)   
## mod.lmer.null2 3 156.374 170.454 -75.187 150.374   
## mod.lmer.null3 3 103.492 117.572 -48.746 97.492 52.882 0   
## mod.lmer.null1 4 69.453 88.226 -30.727 61.453 36.039 1 1.934e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The first null model is the best because it has the lowest AIC and lowest BIC.

mod.lmer.a <- lmer(loginv~logpop+(1|family.name)+(1|area), data=df) #This is the code for a varying intercept model  
  
  
mod.lmer.ab <- lmer(loginv~logpop+(1+logpop|family.name) +(1+logpop|area),data=df) #This is the code for a varying intercept and slope model

## boundary (singular) fit: see help('isSingular')

summary(mod.lmer.a)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: loginv ~ logpop + (1 | family.name) + (1 | area)  
## Data: df  
##   
## REML criterion at convergence: 68.7  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.7071 -0.5947 -0.0535 0.5778 3.4460   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## family.name (Intercept) 0.01817 0.1348   
## area (Intercept) 0.05347 0.2312   
## Residual 0.05517 0.2349   
## Number of obs: 807, groups: family.name, 104; area, 7  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 3.416e+00 9.657e-02 6.766e+00 35.374 6.24e-09 \*\*\*  
## logpop 6.982e-03 2.986e-03 8.049e+02 2.338 0.0196 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## logpop -0.266

summary(mod.lmer.ab)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: loginv ~ logpop + (1 + logpop | family.name) + (1 + logpop |   
## area)  
## Data: df  
##   
## REML criterion at convergence: 68.7  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.7073 -0.5931 -0.0541 0.5811 3.4422   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## family.name (Intercept) 1.763e-02 0.1327610   
## logpop 9.306e-08 0.0003051 1.00   
## area (Intercept) 5.621e-02 0.2370947   
## logpop 4.654e-07 0.0006822 -1.00  
## Residual 5.514e-02 0.2348128   
## Number of obs: 807, groups: family.name, 104; area, 7  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 3.416e+00 9.874e-02 5.815e+00 34.594 5.88e-08 \*\*\*  
## logpop 7.132e-03 2.994e-03 2.335e+02 2.382 0.018 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## logpop -0.342  
## optimizer (nloptwrap) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

So the correlation is still weakly statistically significant.

anova(mod.lmer.null1, mod.lmer.a, mod.lmer.ab)

## refitting model(s) with ML (instead of REML)

## Data: df  
## Models:  
## mod.lmer.null1: loginv ~ (1 | family.name) + (1 | area)  
## mod.lmer.a: loginv ~ logpop + (1 | family.name) + (1 | area)  
## mod.lmer.ab: loginv ~ logpop + (1 + logpop | family.name) + (1 + logpop | area)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)   
## mod.lmer.null1 4 69.453 88.226 -30.727 61.453   
## mod.lmer.a 5 65.927 89.394 -27.964 55.927 5.5259 1 0.01874 \*  
## mod.lmer.ab 9 73.874 116.114 -27.937 55.874 0.0527 4 0.99966   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

According to the BIC the null model accounts for the variation better than the models with the predictor variables.

## Coefficients from a multilevel model

There isn’t just a single coefficient for a multilevel model

beta1 <- coef(mod.lmer.ab)$family.name  
colnames(beta1) <- c("Intercept", "Slope")  
  
beta2 <- coef(mod.lmer.ab)$area  
colnames(beta2) <- c("Intercept", "Slope")

If we plot the coefficients we can see that they follow a normal distribution.

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

p1 <- ggplot(beta1, aes(Slope))+  
 geom\_density(fill="slategray2", color="slategray2", alpha=0.8)+  
 ggtitle("Random slopes by linguistic family")  
  
  
p2 <- ggplot(beta1, aes(Intercept))+  
 geom\_density(fill="slategray2", color="slategray2", alpha=0.8)+  
 ggtitle("Random intercepts by linguistic family")  
  
p3 <- ggplot(beta2, aes(Slope))+  
 geom\_density(fill="slategray2", color="slategray2", alpha=0.8)+  
 ggtitle("Random slopes by area")  
  
p4 <- ggplot(beta2, aes(Intercept))+  
 geom\_density(fill="slategray2", color="slategray2", alpha=0.8)+  
 ggtitle("Random intercepts by area")  
  
grid.arrange(p1, p2, p3, p4, ncol=2)

