HW6 Linear mixed-effect models

0. Packages and data

```
library(tidyverse)
library(lme4)
library(lmerTest)
library(lingtypology) # only for linguistic mapping
library(broom)
```

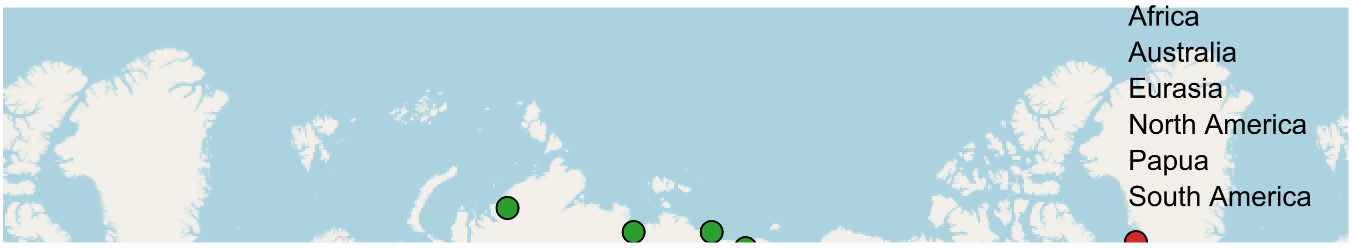
UPSID database

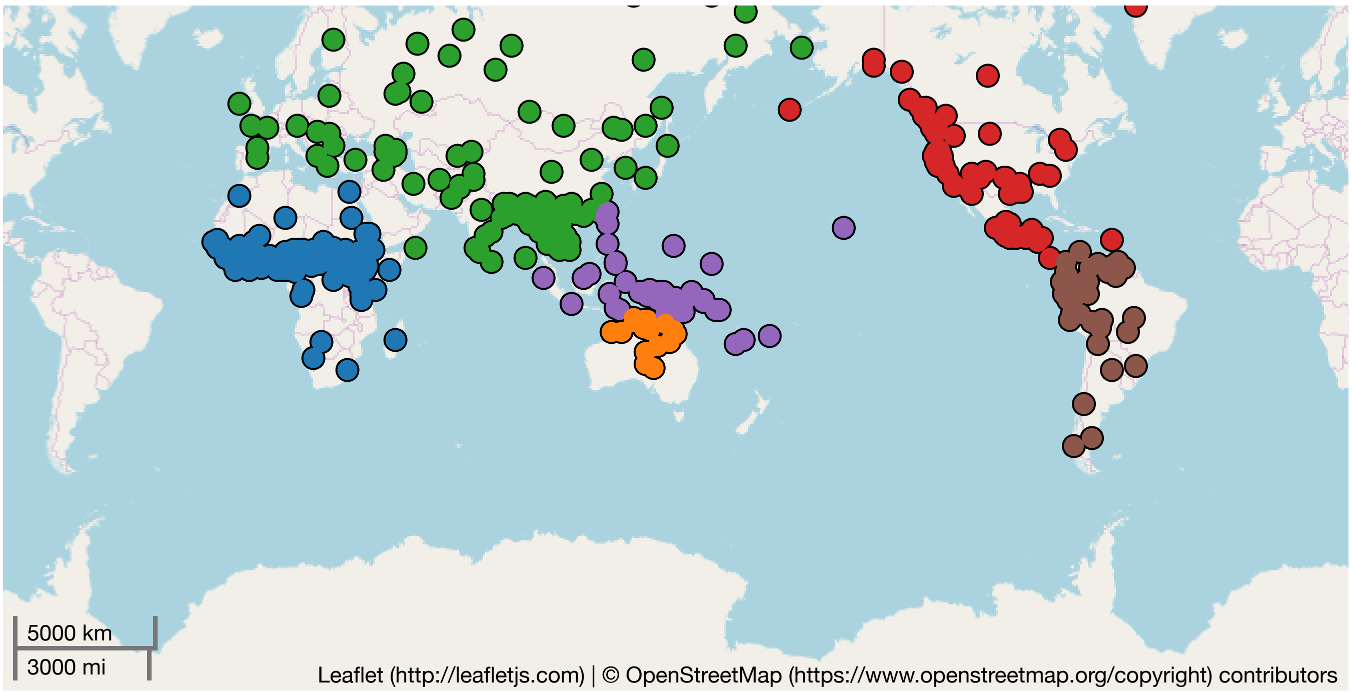
In this dataset we have number of consonants and vowels in 402 languages collected from UPSID database (http://www.lapsyd.ddl.ish-lyon.cnrs.fr/lapsyd/ (http://www.lapsyd.ddl.ish-lyon.cnrs.fr/lapsyd/)).
* `language` - language
* `area` - language area according to Glottolog (http://glottolog.org/ (http://glottolog.org/))
* `consonants` - the number of consonants in the language
* `vowels` - the number of vowels in the language

```
upsid <- read_csv("https://raw.githubusercontent.com/LingData2019/LingData2020/maste
r/data/upsid.csv")
upsid
```

language <chr>	area <chr>	consonants <dbl>	vowels <dbl>
Afade	Africa	27	9
Achumawi	North America	17	6
Adzera	Papua	18	7
Eastern Arrernte	Australia	27	3
Amele	Papua	15	5
Angaataha	Papua	12	9
Aghem	Africa	25	10
Aproumu Aizi	Africa	24	9
Ahtena	North America	30	5
Hokkaido Ainu	Eurasia	11	5

1-10 of 451 rows   Previous **1** 2 3 4 5 6 ... 46 Next




5000 km
3000 mi
Leaflet (http://leafletjs.com) | © OpenStreetMap (https://www.openstreetmap.org/copyright) contributors

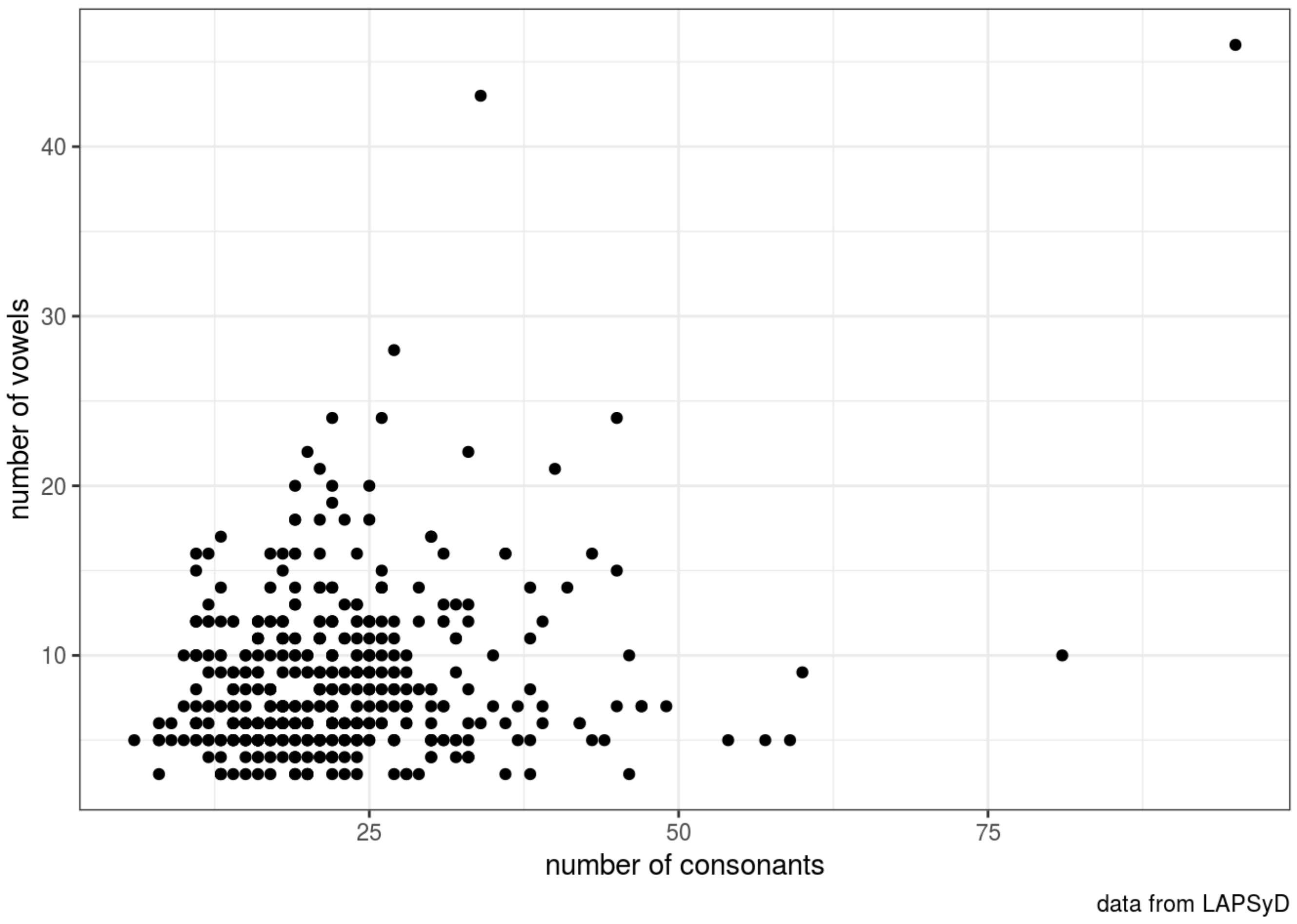
```
# you can map the languages using the lingtypology package
# map.feature(upsid$language,
#             features = upsid$area,
#             label = upsid$language,
#             label.hide = TRUE)
```

In this work, you will fit a number of linear and linear mixed-effect models that predict the number of vowels in a language by the number of consonants and other variables.

1. Linear model

1.1

Make a scatterplot of the number of consonants and the number of vowels using the `ggplot()`.



```
# we use the theme_bw() theme, you can use other themes if you want
```

1.2

Fit the basic linear model `fit1` that predicts the number of vowels by the number of consonants. Look at the summary of the model.

1.3

Is `consonants` a significant predictor? Write down YES or NO.

1.4

no evaluation
To draw predictions on the graph, we may use `fortify` function. It adds column entitled `.fitted` (note period) that is calculated as a prediction of a model as well as some other columns.

Try this code and see the result:

```
head(fortify(fit1))
```

	vow… <dbl>	consonants <dbl>	.hat <dbl>	.sigma <dbl>	.cooksd <dbl>	.fitted <dbl>	.resid <dbl>	.st
1	9	27	0.002722938	4.675111	5.283118e-09	9.009174	-0.009174129	-0.0019

	vow...	consonants	.hat	.sigma	.cooksd	.fitted	.resid	.st
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
2	6	17	0.002953279	4.674262	2.415722e-04	7.883257	-1.883256653	-0.4038
3	7	18	0.002708692	4.674874	6.192365e-05	7.995848	-0.995848401	-0.2135
4	3	27	0.002722938	4.666459	2.266680e-03	9.009174	-6.009174129	-1.2885
5	5	15	0.003590156	4.673418	5.857681e-04	7.658073	-2.658073158	-0.5702
6	9	12	0.004914727	4.674434	3.210686e-04	7.320298	1.679702085	0.3605

6 rows

Another option is to use `augment` function from `broom` library that works in a similar way.

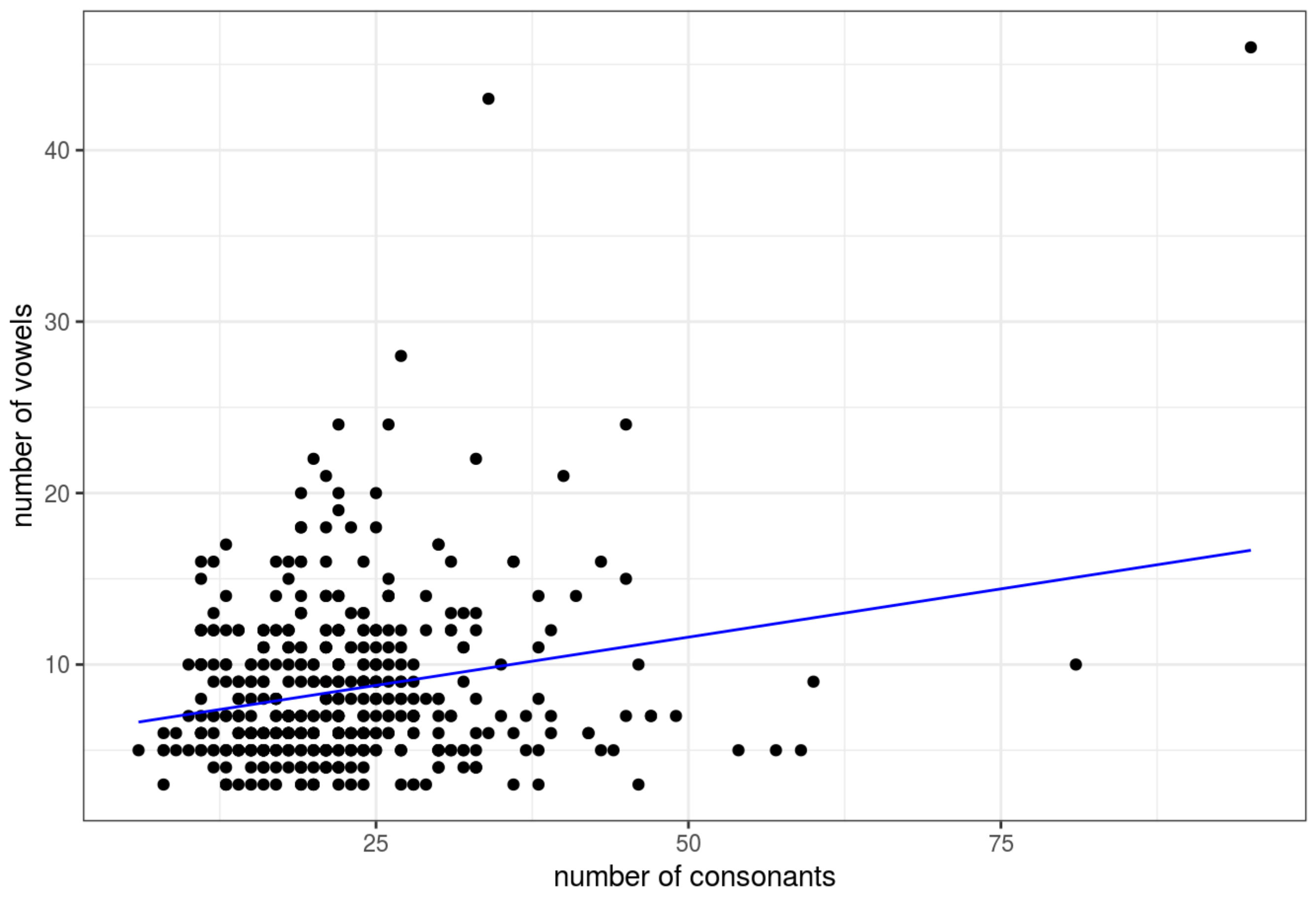
```
library(broom)
head(augment(fit1))
```

vow...	consonants	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
9	27	9.009174	0.2436838	-0.009174129	0.002722938	4.675111	5.283118e-09
6	17	7.883257	0.2537815	-1.883256653	0.002953279	4.674262	2.415722e-04
7	18	7.995848	0.2430455	-0.995848401	0.002708692	4.674874	6.192365e-05
3	27	9.009174	0.2436838	-6.009174129	0.002722938	4.666459	2.266680e-03
5	15	7.658073	0.2798108	-2.658073158	0.003590156	4.673418	5.857681e-04
9	12	7.320298	0.3273840	1.679702085	0.004914727	4.674434	3.210686e-04

6 rows

1.5

Use one of these function to add a line to the scatterplot 1.1. You have to use `geom_line` and pass the result of `augment` or `fortify` as argument `data` in `geom_line`. (E.g. `geom_line(data = <result of augment>, ...`))



data from LAPSyD

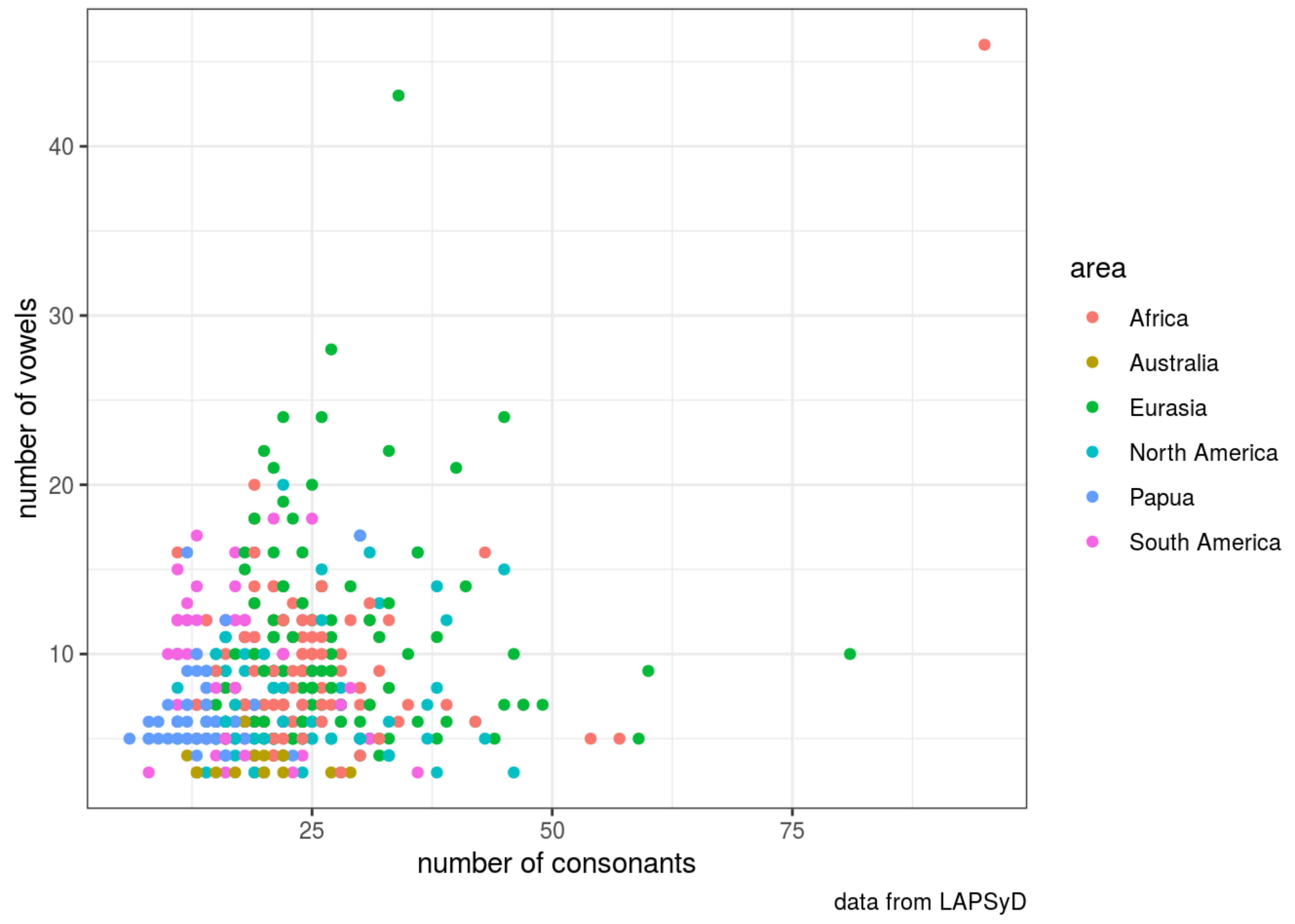
```
#  geom_line(data = augment(fit1), aes(x = consonants, y = .fitted), color = "blue")
```

2. Mixed-effect models

Let us look at the data with respect to the `area` groups.

2.1

Re-build the scatterplot `1.1` coloring the points by `area`.



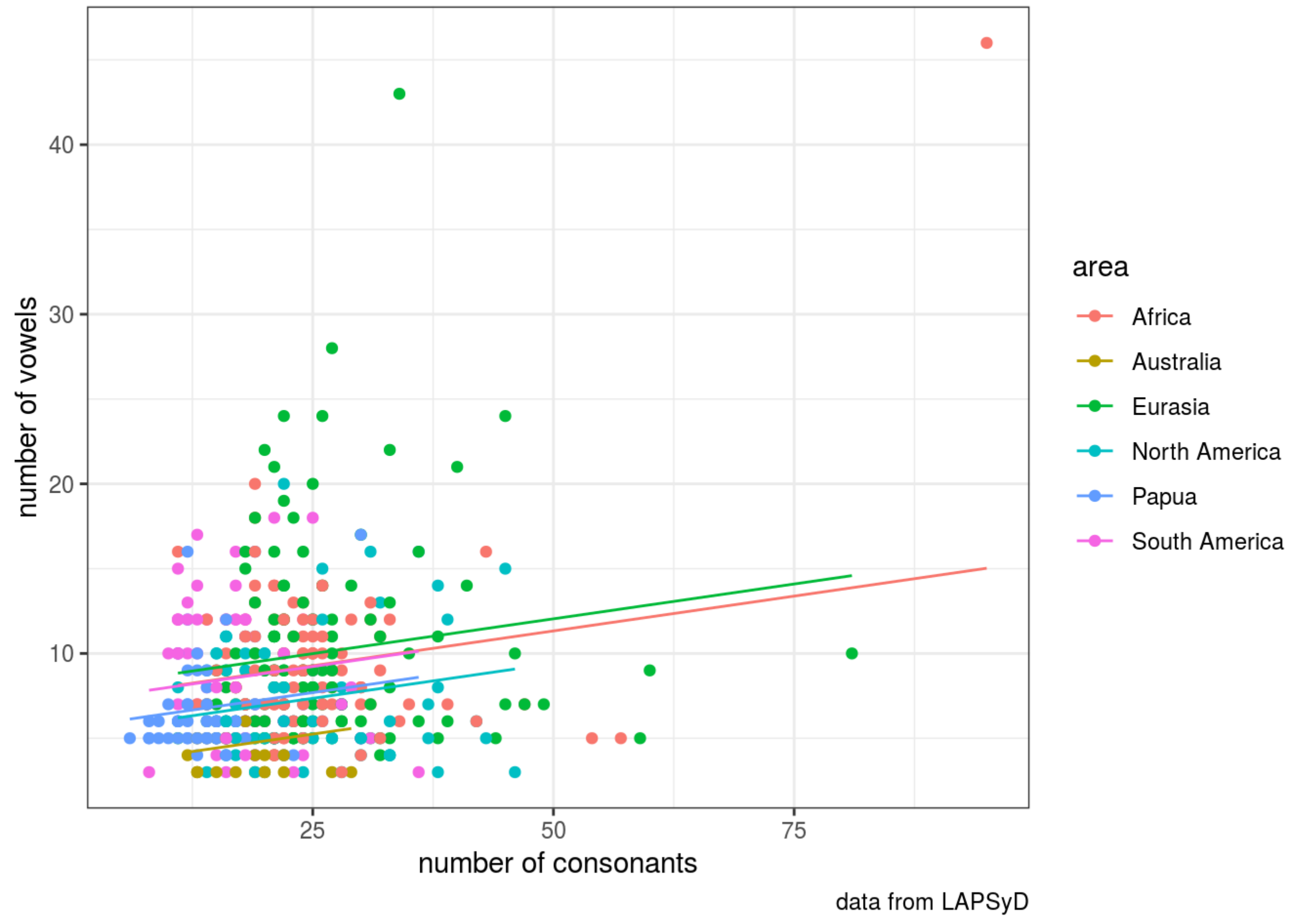
2.2

Use lmer() to fit the model `fit2` with random `area` group intercept. Your model is given by equation:

$$vowels = \beta_0 + \beta_1 \times consonants + u(area).$$

2.3

Add the regression lines to the scatterplot `2.1` using `fortify` or `augment` methods.



2.4

Interpret the results of the model. (Use `summary()` to get information about your model.) Is *consonansts* variable still significant? How it changes after we added *area* random effect? What can you say about lines on the graph? How are they located with respect to each other? Why?

3. Mixed-effect models with random slopes.

3.1

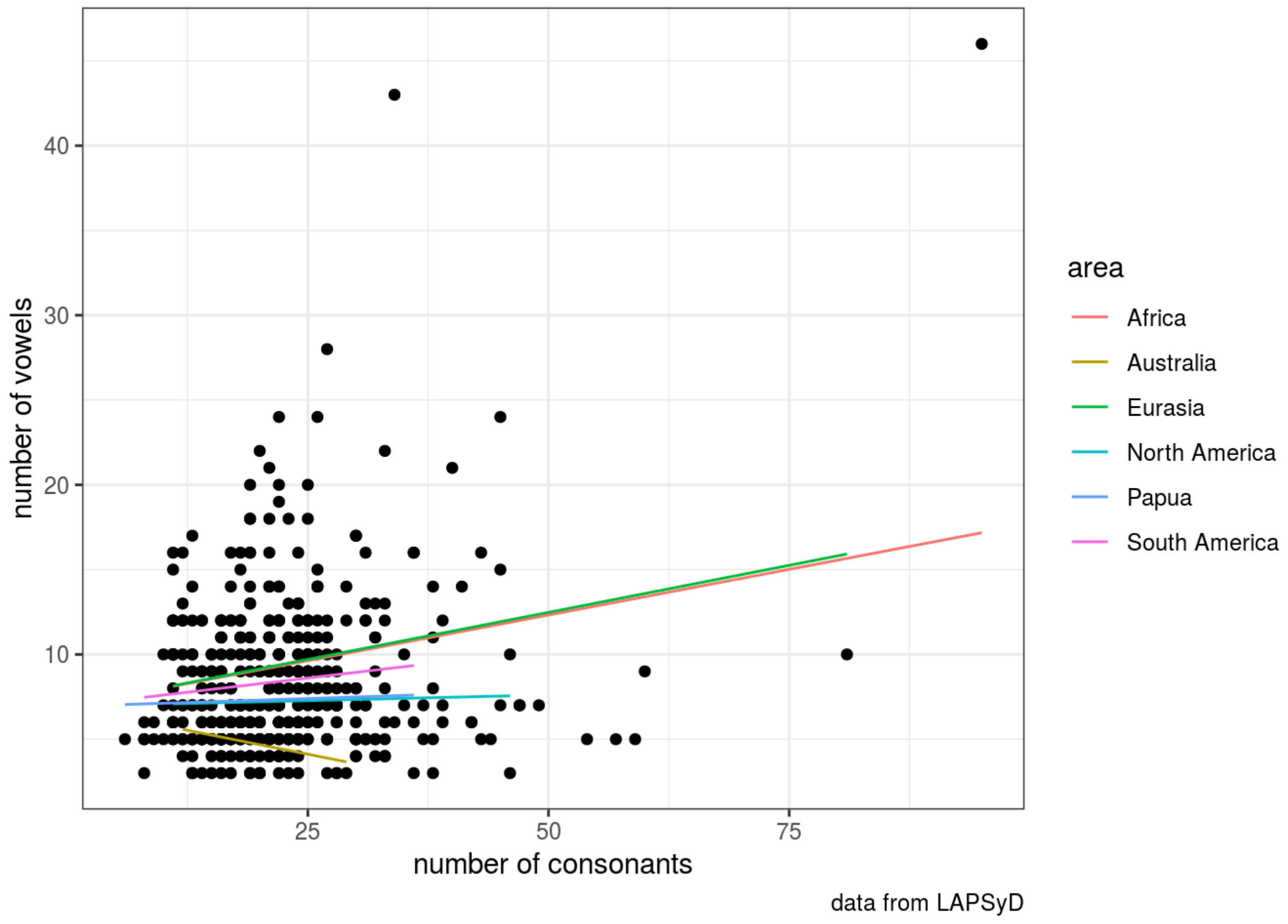
Fit the model `fit3` with random slope that depends on *area*. Your model is given by formula:

$$vowels = \beta_0 + (\beta_1 + u(area)) \times consonants.$$

You can consult this (http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#model-specification) manual to choose the correct syntax in your `lmer` formula.

3.2

Draw a figure with prediction lines:



3.3

Interpret the results. Is *constants* still significant? Why? What can you say about lines on the graph? How are they located with respect to each other? Why?

4. Mixed-effect models with random intercept and random slopes.

Now let us assume we have both random intercept and random slope (and they are not correlated). Our model is of the form:

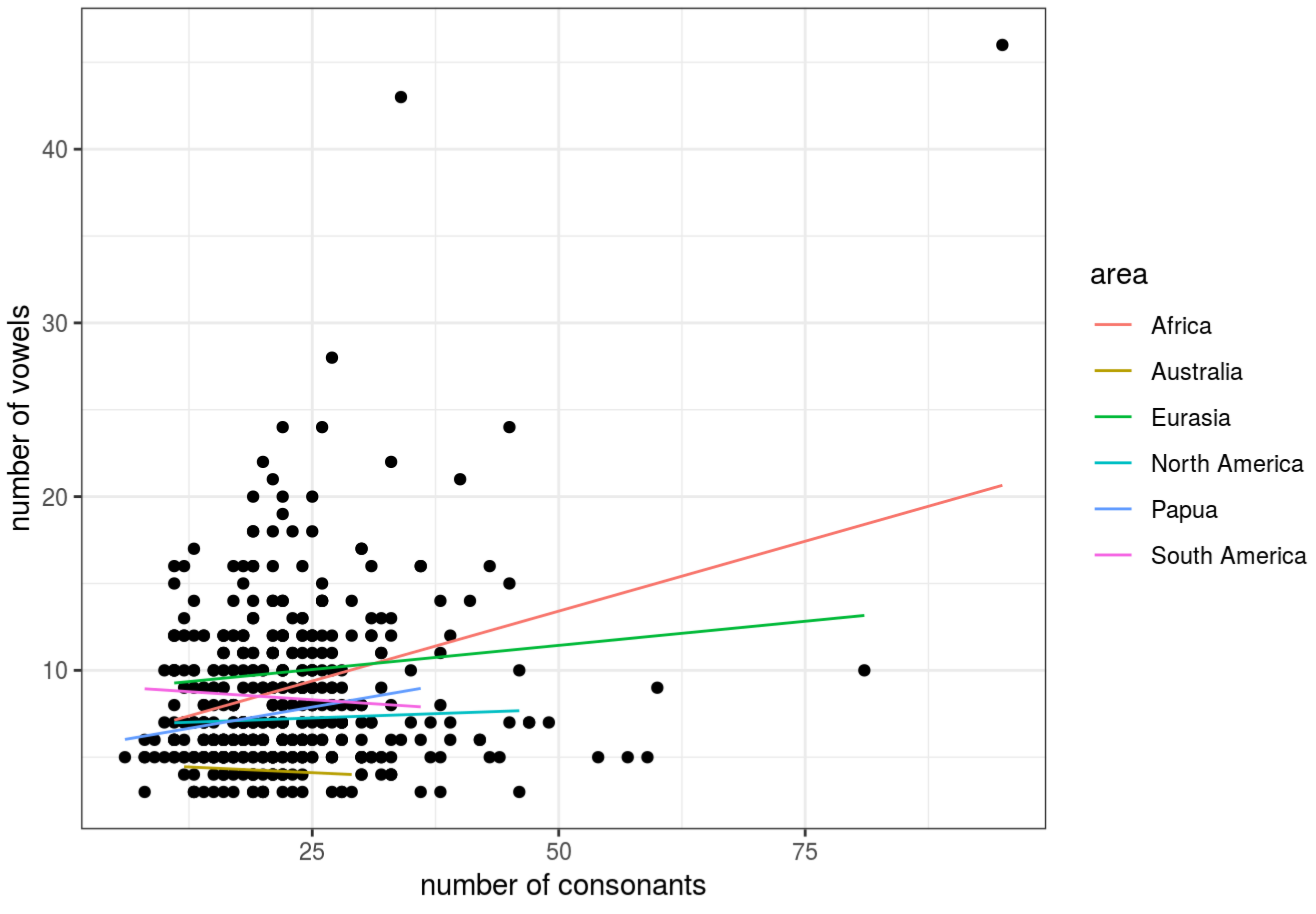
$$vowels = \beta_0 + u_1(area) + (\beta_1 + u_2(area)) \times consonants.$$

4.1

Fit the model `fit4`.

4.2

Draw a figure with predictions:



data from LAPSyD

4.3

Interpret the results. Is *constants* significant? How can we interpret the difference between the prediction lines of these three models?