GEOG 714 - Assignment 6

Haoran Xu

2024-11-03

```
x1 <- rnorm(300)
groups <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
g1 <- sample(groups, 300, replace=TRUE)
df <- data.frame(x1, g1)
df$y <- 5 * x1 + rnorm(300)
```

Q1. Use lm to analyze the data. How might you incorporate g1 into this modelling process if you had to?

```
lm(df$y ~ df$x1)
lmout1 <- lm(df$y ~ df$x1)
summary(lmout1)
```

The results of OLS two-varibale linear regression showed that y is significantly and positively correlated with x1 (p-value < 0.001), with x1 increasing each 1, y1 is estimated to increase 5.013. And the Adjusted R-squared (0.965) indicated that the model fits well. If I would incorporate g1, there would be two ways. One is to using the calculated mean within each group as indepent variables to be used in linear regression. This way would try to incorporate the information of all observations but may lose the dicrepancies within each group, and also this would greatly lower the number of variables from 300 to 10. Another way is to conducting separate linear regression models for each group. In this case we would have to do lm for 10 times. This method can make the results "noisy" in that the estimates from each model are not based on very much data.

```
group_effect1 <- rnorm(10)
df$g1_effect1 <- group_effect1[g1]  # assign the grouping effect using index of the group, that's why t
```

Q2. In one or two sentences, explain what is happening in this code.

The code first created a vector of 10 random numbers (in normal distribution with a mean of 0 and standard deviation of 1) representing the varied effects for each group. Then the `g1_effect` variable was added to `df` by assigning the `group_effect` value to each observation using the index of the `group_effect` vector (also the group number).

```
df$yre <- 5 * x1 + df$g1_effect1 + rnorm(300)
```

Q3. Write code that will create a unique regression model of the association between x1 and yre for each group. This should result in 10 models. Write code to store the results of each model in separate files (out1, out2, ..., out10). You may wish to consider using a for loop for this task, and saving the output from the models in a single list() data structure.

```
df_list <- list()
out <- list()

for (i in 1:10) {
  df_list[[i]] <- df[df$g1 == i, ]
  out[[i]] <- lm(yre ~ x1, data = df_list[[i]])
  saveRDS(out[[i]], paste0("out", i, ".rds"))
}

out
```

```
group_effect2 <- rnorm(10, 0, 10)
df$g1_effect2 <- group_effect2[g1]
df$yre <- 5 * x1 + df$g1_effect2 + rnorm(300)
lme4_out <- lmer(yre ~ x1 + (1|g1), data = df)
summary(lme4_out)

re <- ranef(lme4_out)$g1 # create a vector caputring the different residuls for each group
plot(re$`(Intercept)`, group_effect2)   # should use `(Intercept)` to represent the x variable
```

Q4. How would you judge the success of the model at estimating the random effects associated with the groups based on looking at the plot? Explain your reasoning.

I would check the correlation between the model's random intercepts (`re$'(Intercept)'`) and the pre-assigned group effects (`group_effect2`) in theplot. If the model is successful, the relationships between the two varibles would be significantly and positively correlated, while also aligning with the the `y = x` line. The plot has shown this pattern, suggesting the model is pretty successful.

```
df2 <- read.csv(paste0(here(), "/Assignments/Assignment6/AmesHousing.csv"))
summary(lm(df2$SalePrice ~ df2$Gr.Liv.Area))
```

Q5. Interpret the results of this model. What is the impact of living area (in square feet) on the value of a home?

The model shows that living area has a significant and positive impact on home value. Specifically, for each additional square foot of living area, the sale price of a home was estimated to increases by 111.69. The high p-value (< 2e-16) indicated that this effect is statistically significant. The adjusted r-squared value of 0.4995 suggests that around 50% of the variability in home sale prices can be explained by the living area.

```
df2$Neighborhood <- factor(df2$Neighborhood)
df2$SalePrice <- scale(df2$SalePrice)  # Normal Standardization (this would change the name of the colu
df2$Gr.Liv.Area <- scale(df2$Gr.Liv.Area)
```

```r
out_SalePrice1 <- lm(SalePrice ~ Gr.Liv.Area, data = df2)
summary(out_SalePrice1)

out_SalePrice2 <- lmer(SalePrice ~ (1|Neighborhood), data = df2)
summary(out_SalePrice2)

out_SalePrice3 <- lmer(SalePrice ~ Gr.Liv.Area + (1 + Gr.Liv.Area|Neighborhood), data = df2)
summary(out_SalePrice3)
```

```r
N <- as.character(unique(df2$Neighborhood))

df2$predicted <- predict(out_SalePrice3)

for (i in 1:28){
  if(i == 1){
    plot(
      df2$Gr.Liv.Area[df2$Neighborhood == N[i]],
      df2$predicted[df2$Neighborhood == N[i]],
      ylim = c(-4, 4), pch = 20, cex = 0.5, col = "red",
      ylab = "Predictions of sales price", xlab = "Living space"
    )
  }
  points(
    df2$Gr.Liv.Area[df2$Neighborhood == N[i]],
    df2$predicted[df2$Neighborhood == N[i]],
    pch = 20, cex = 0.5, col = "blue"
  )
}
```

Q6. What is the 'N' vector, and what is it doing in this code?

The `N` vector contains the unique neighborhood names from the `df2$Neighborhood`. And this code is using loop to plot the relationships between living space (normally standized) and predictions of sale price for each neighhood.

Q7. Use another independent variable in this data set and go through the same steps above, interpreting the results accordingly.

```r
df2$Bldg.Type <- factor(df2$Bldg.Type)

unique(df2$Bldg.Type)

out_SalePrice4 <- lmer(SalePrice ~ Gr.Liv.Area + (1 + Gr.Liv.Area|Bldg.Type), data = df2)
```

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

```r
summary(out_SalePrice4)

N <- as.character(unique(df2$Bldg.Type))
```

```r
df2$predicted2 <- predict(out_SalePrice4)

for (i in 1:5){
  if(i == 1){
    plot(
      df2$Gr.Liv.Area[df2$Bldg.Type == N[i]],
      df2$predicted2[df2$Bldg.Type == N[i]],
      ylim = c(-4, 6), pch = 20, cex = 0.5, col = "red",
      ylab = "Predictions of sales price", xlab = "Living space"
    )
  }
  points(
    df2$Gr.Liv.Area[df2$Bldg.Type == N[i]],
    df2$predicted2[df2$Bldg.Type == N[i]],
    pch = 20, cex = 0.5, col = "green"
  )
  max_x <- max(df2$Gr.Liv.Area[df2$Bldg.Type == N[i]])
  max_y <- max(df2$predicted2[df2$Bldg.Type == N[i]])
  text(max_x, max_y, labels = N[i], pos = 2, cex = 0.5, col = "blue")
}
```

I chose `Bldg.Type` as the grouping varibale. This factorial variable has 5 groups - "1Far, 2fmCon, Duplex, Twnhs, TwnhsE", which may represent "Single-family detached house, Two-family conversion, Duplex, Townhouse End Unit, Townhouse Inside Unit".

The results showed that 1) generally, the predicated sale prices increase positively (0.562) and significantly (t-value > 3.6) with the living space getting bigger, according to the fixed effects; 2) sale prices varie by building type (the variance of random intercept is $0.481 > 0$); 3) the association between sale prices and living space varies systematically across building types, with the random slope effect > 0; 4) in building types with higher sales prices (such as Single-family detached house or Townhouse Inside Unit), the relationship between sales price and living space is stronger (the slope is steeper) than in those with lower sales prices (as the correlation between random intercept and slope effects is 1.00), as also shown in the graphics.