Introduction to R and the tidyverse

Practice exercises

Kim Dill-McFarland, kadm@uw.edu

version July 07, 2020

Contents

Overview

These exercises will help you to practice tidyverse and other functions covered in the Intro R workshop including:

- Subsetting
- Pivoting
- Joining
- Plotting

Setup

There was an error in the RNAseq results we used in the workshop. Please download the updated file from the workshop Dropbox.

Open the Intro R Rproject and start a new working script. Load packages, set a seed, and load data as we did in the workshop.

```
library(tidyverse)
library(limma)
library(broom)
#Set seed
set.seed(4389)

#SNP genotypes
snp <- read_csv(file="data/Hawn_RSTR_SNPlist.PRKAG2.csv")
#RNAseq expression and metadata
load("data/RSTR_RNAseq_dat.voom_updated.RData")
```

Exercises

Subsetting

Subset the `snp` data frame to the rows and/or columns you need to answer the following questions.

1. How many SNPs were annotated to type "exon"?
2. What is the genotype of donor 89337-1-06 for SNP rs77961133?
3. What are the maximum and minimum positions (POS) of SNPs in PRKAG2?
4. Challenge: What is the snpID of the SNP that was annotated to a "promoter"? Hint: promoter is only one of the annotations for this SNP. Functions such as `grepl( )` allow you to search for patterns within a value instead of exact matches as with `==`.

Pivoting

1. Without running the following code, sketch the resulting data frame structures. Once you've done this, check the outputs in R.

```
snp %>%
  filter(type == "intron, exon") %>%
  select(rsID, `91053-1-04`, `84222-1-19`) %>%
  pivot_longer(-rsID, names_to="FULLIDNO", values_to="genotype")
```

```
snp %>%
  select(snpID, type, POS) %>%
  pivot_wider(names_from = type, values_from = POS)
```

2. Challenge: Instead of converting 0/0 formatted genotypes to numeric 0,1,2 (as we did in the workshop), convert them to their alleles such as A/T. Below is a skeleton of the workflow with blanks indicated as `[SOMETHING]`.

```
geno <- snp %>%
  #Select genotype data
  select( [SOMETHING] ) %>%
  #Convert to long format
  pivot_longer( [SOMETHING],
                names_to="FULLIDNO",
                values_to="genotype") %>%
  #Convert genotype to alleles
  mutate(geno.allele = ifelse(genotype == "0/0",
                              paste(allele.0, allele.0, sep="/"),
                       ifelse(genotype == "0/1",
                              [SOMETHING],
                       ifelse(genotype == "1/1",
                              [SOMETHING],
                              NA)))) %>%
  #Convert back to wide format
  select(snpID, FULLIDNO, geno.allele) %>%
  pivot_wider(names_from = [SOMETHING],
              values_from = [SOMETHING])
```

Joining

1. Using the gene information in `dat.norm.voom$genes` and a join function, relabel the genes in the expression data `dat.norm.voom$E` with ENSEMBL ID.
2. Use the following code to create two new data frames.

```
df1 <- data.frame(donor = c("A","B","C"),
                  age = c(10,14,4))

df2 <- data.frame(donor = c("A","C","D"),
                  sex = c("F","M","F"))
```

Then sketch what the resulting data frames would be from the following join functions before running them in R.

```
left_join(df1, df2, by = "donor")

right_join(df1, df2, by = "donor")

full_join(df1, df2, by = "donor")
```

Plotting

Using the metadata in `dat.norm.voom$targets`,

1. Create a dot plot of BMI by age.
   - Remember that there are 2 rows for each donor (one for MEDIA and one for TB samples) so to best plot these data, you should filter to one age/BMI value per person.
2. Why is there a warning message for missing values when you make the age/BMI plot?
3. Does there appear to be a linear relationship between age and BMI? Check by adding a fit line to the plot and running a linear model. Note that you can pipe directly into `lm( )` like so

```
data.frame %>%
  lm(y ~ x , data=.) %>%
  tidy()
```

4. Using facets, check if age, BMI, or risk score differ between LTBI and RSTR groups. Below is a skeleton of the workflow with blanks indicated as `[SOMETHING]`.

```
dat.norm.voom$targets %>%
  #Keep 1 row for each donor
  filter( [SOMETHING] ) %>%
  #Select variables of interest
  select(sampID, Sample_Group, M0_KCVAGE, avgBMI, RISK_SCORE) %>%
  #Long format
  pivot_longer(M0_KCVAGE:RISK_SCORE) %>%

  #Boxplot
  ggplot(aes(x = [SOMETHING], y = [SOMETHING] ))+
  geom_boxplot() +
  #Facets
  #Try removing the scales option to see what it does!
  facet_wrap(~ [SOMETHING], scales = "free") +
  theme_classic()
```

R session

```
sessionInfo()
```

```
## R version 4.0.0 (2020-04-24)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.0.0  magrittr_1.5    tools_4.0.0     htmltools_0.5.0
##  [5] yaml_2.2.1      stringi_1.4.6   rmarkdown_2.3   knitr_1.29
##  [9] stringr_1.4.0   xfun_0.15       digest_0.6.25   rlang_0.4.6
## [13] evaluate_0.14
```