Mendel Chi-square Workshop Introduction to R Workshop

The R Coding Language and RStudio IDE

R is a popular open-source coding language used by scientists worldwide for statistical analysis and data visualization. The best and most popular way to use R is with RStudio: an integrated development environment (IDE) that allows users to write and run code while visualizing outputs and keeping track of objects and files written code utilizes and creates.

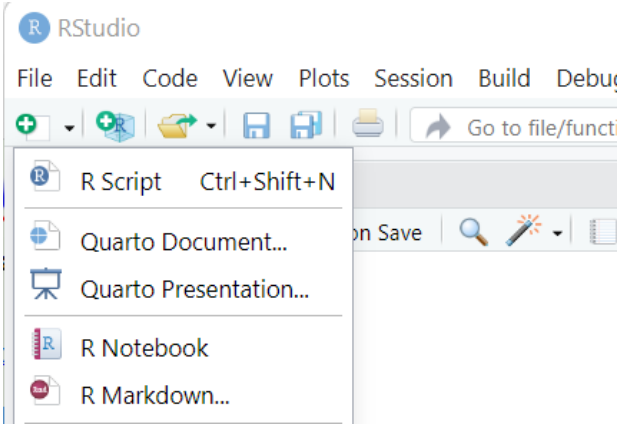
Downloading and Installing R and RStudio

The process of downloading and installing R varies depending on your type of computer. Mac users must ensure they download the correct version for the Mac OS on their computer and must also download and install XQuartz.

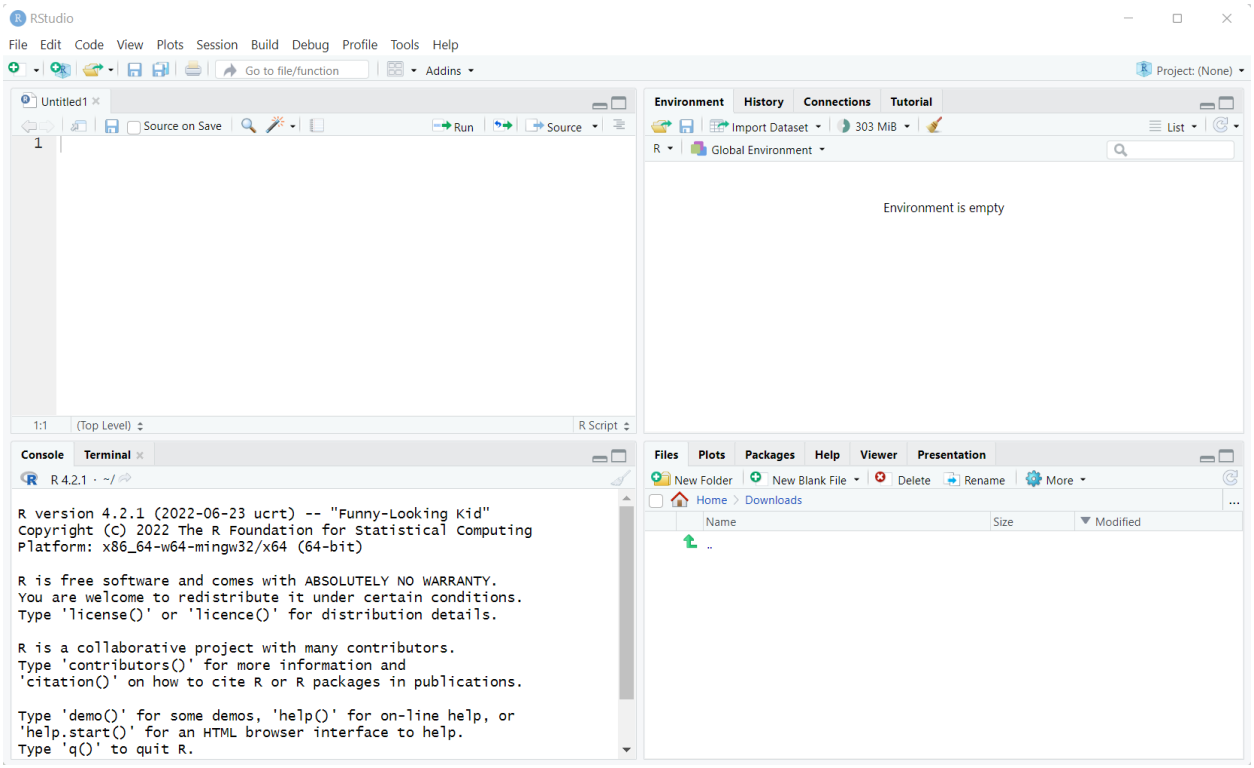
The RStudio Download Website will take you through this process, but please feel free to email Patrick or ask for help in class!

Getting Familiar with RStudio and Reading in Data!

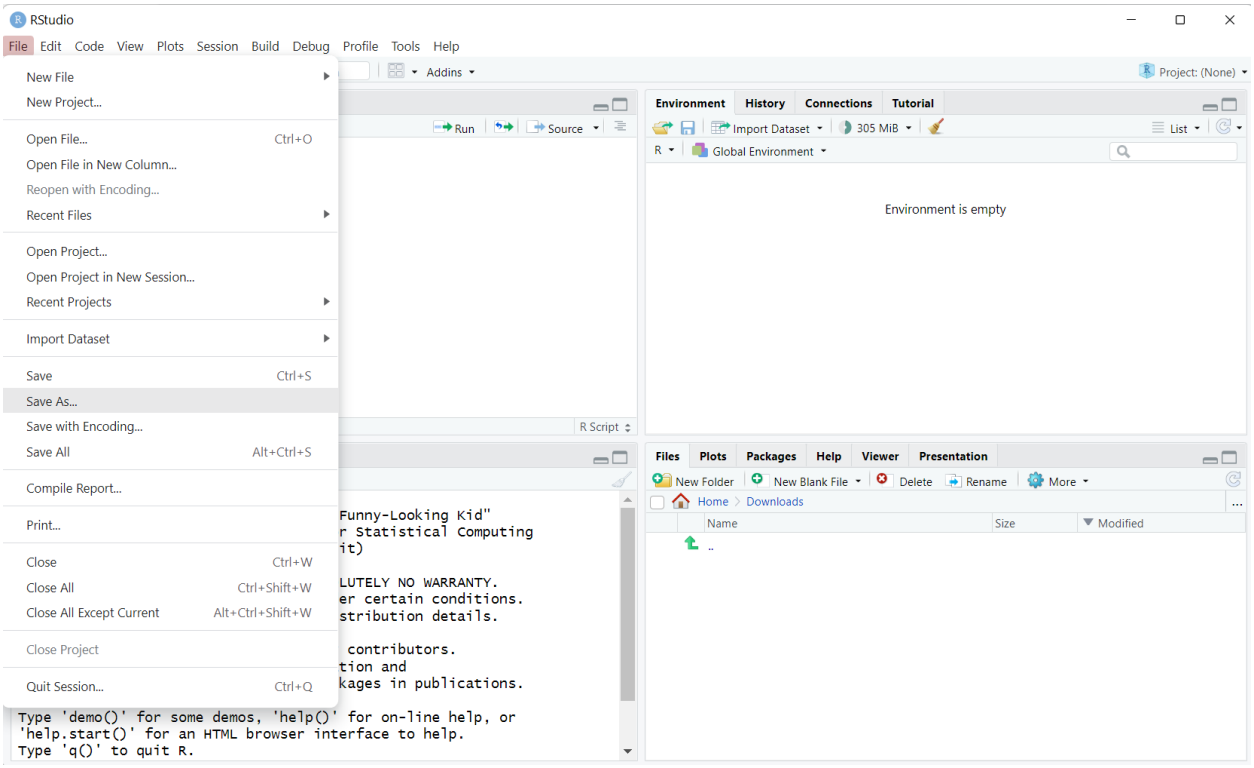
After opening RStudio, click the small white box with the green plus sign to show the drop-down menu for new files, and click 'R Script' to open your first .R file!

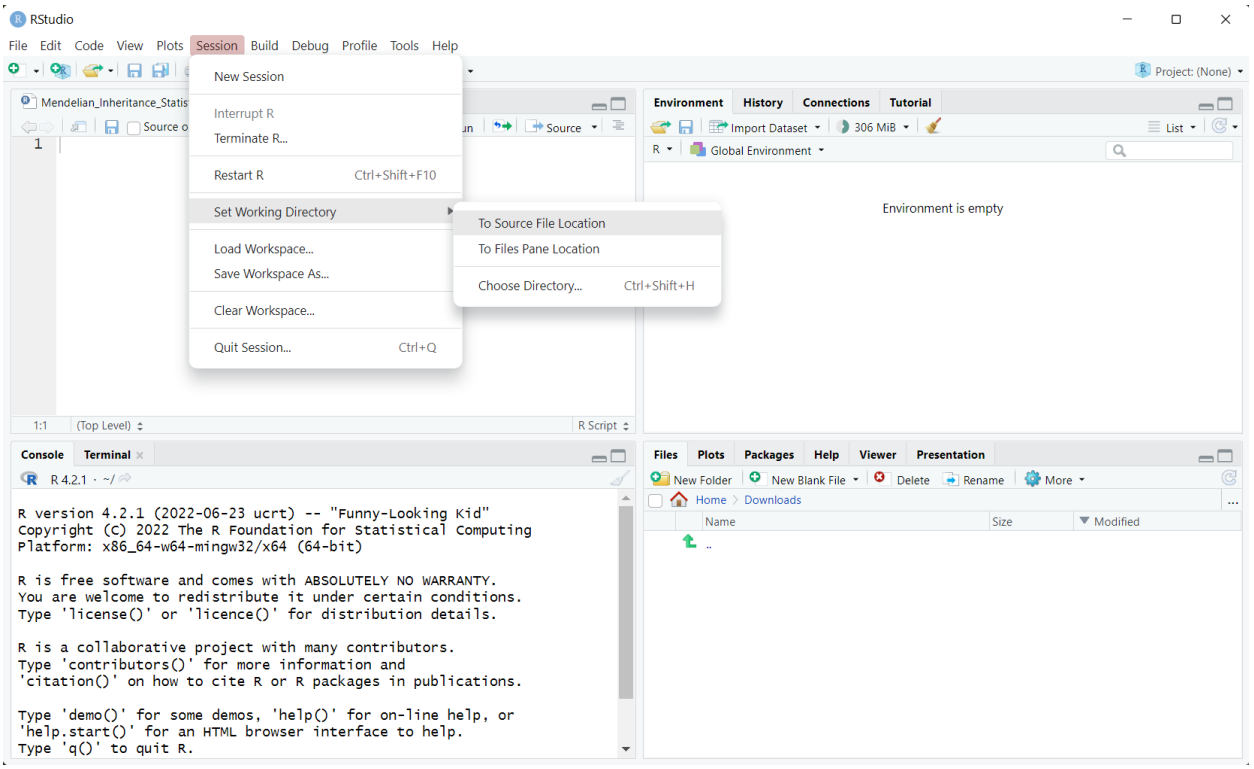


Your RStudio should now look something like the image below with four panels. The upper left panel displays your .R script file where you will write code. The upper right panel shows your 'Environment' where objects you read into R and create will be displayed. The bottom right panel displays your Console, where your code will be run after it is written in the above panel. The bottom left panel shows your computer's files on the 'Files' tab (similar to Finder or File Explorer), graphics that your code creates on the 'Plots' tab, the 'packages' or additional software you may use on the 'Packages' tab, and a very helpful searchable help guide on the 'Help' tab

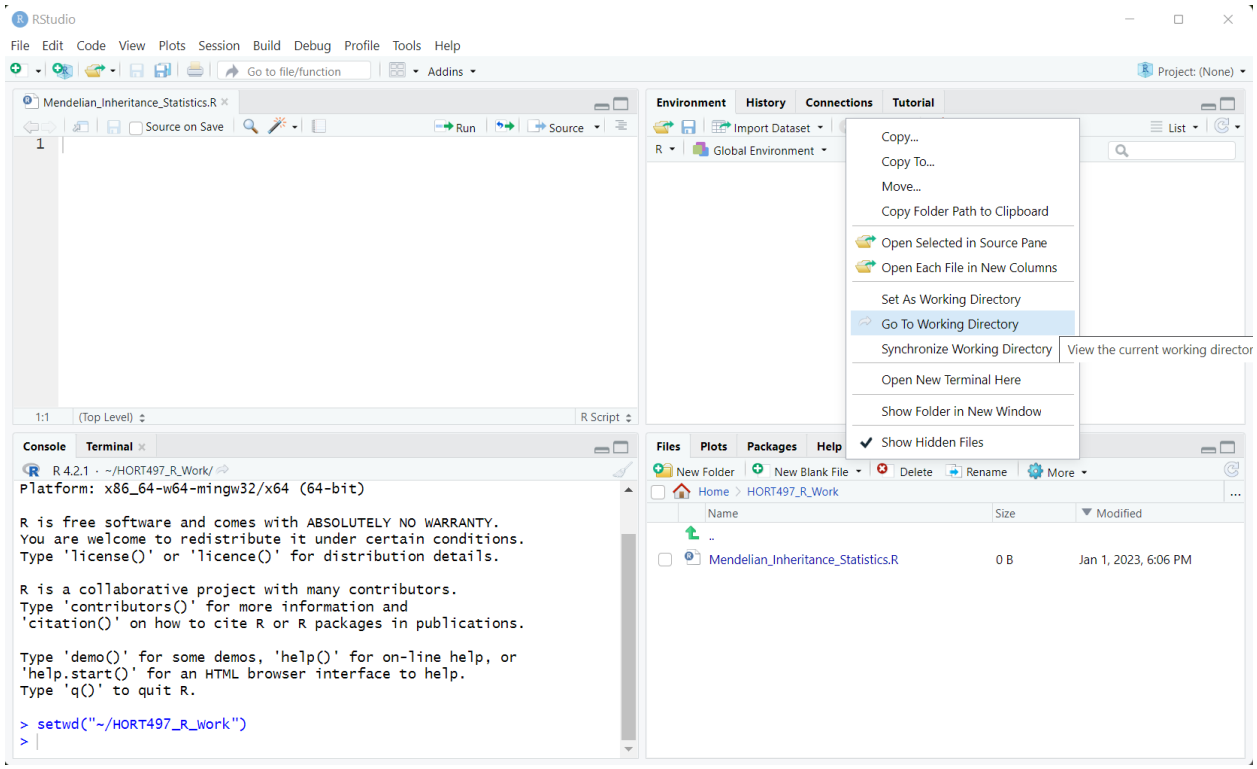


After creating a new file, the next step is to 'set your working directory.' You can think of a working directory as a central location your computer will go to when asked to look for files. The simplest way to do this is by setting your working directory to be the same location where you save your .R code scripts. To do this, create a folder to serve as your working directory and save your .R script in this folder by going to File -> Save As... (NOTICE you must save your file with a name that ends in .R (e.g. "Mendelian_Inheritance_Statistics.R") for the file to be saved and recognized as an .R script file.) Then set your working directory by going to Session -> Set Working Directory -> To Source File Location. After clicking this, you should see that a setwd() command was written and run in the Console panel.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
New File
New Project...
Open File... Ctrl+O
Open File in New Column...
Reopen with Encoding...
Recent Files
Open Project...
Open Project in New Session...
Recent Projects
Import Dataset
Save Ctrl+S
Save As...
Save with Encoding...
Save All Alt+Ctrl+S
Compile Report...
Print...
Close Ctrl+W
Close All Ctrl+Shift+W
Close All Except Current Alt+Ctrl+Shift+W
Close Project
Quit Session... Ctrl+Q
Environment History Connections Tutorial
Environment is empty
Files Plots Packages Help Viewer Presentation
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

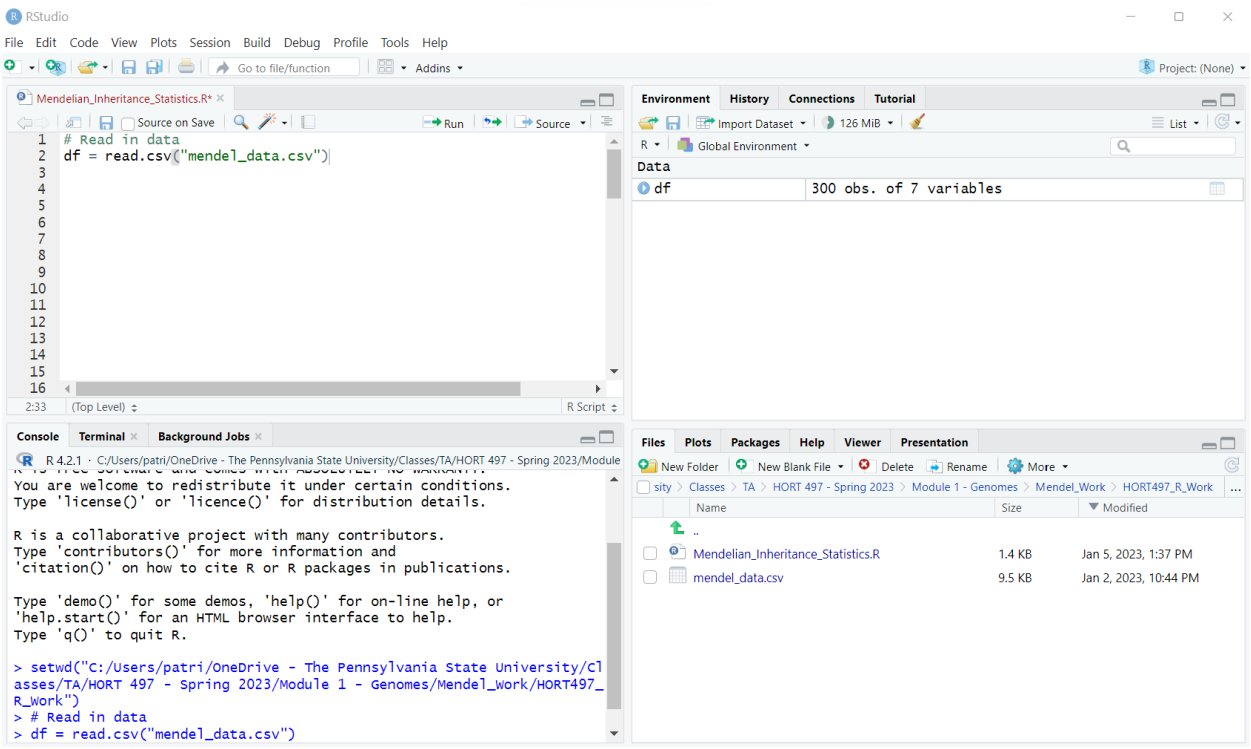

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
New Session
Interrupt R
Terminate R...
Restart R Ctrl+Shift+F10
Set Working Directory
To Source File Location
To Files Pane Location
Choose Directory... Ctrl+Shift+H
Load Workspace...
Save Workspace As...
Clear Workspace...
Quit Session... Ctrl+Q
Environment is empty
Console Terminal
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

To view your working directory and the files within it, hover over the blue gear in the bottom right panel and click "Go To Working Directory". Also, notice how you can also set your working directory by navigating to whatever location you wish in the RStudio 'Files' tab and clicking More -> Set As Working Directory.



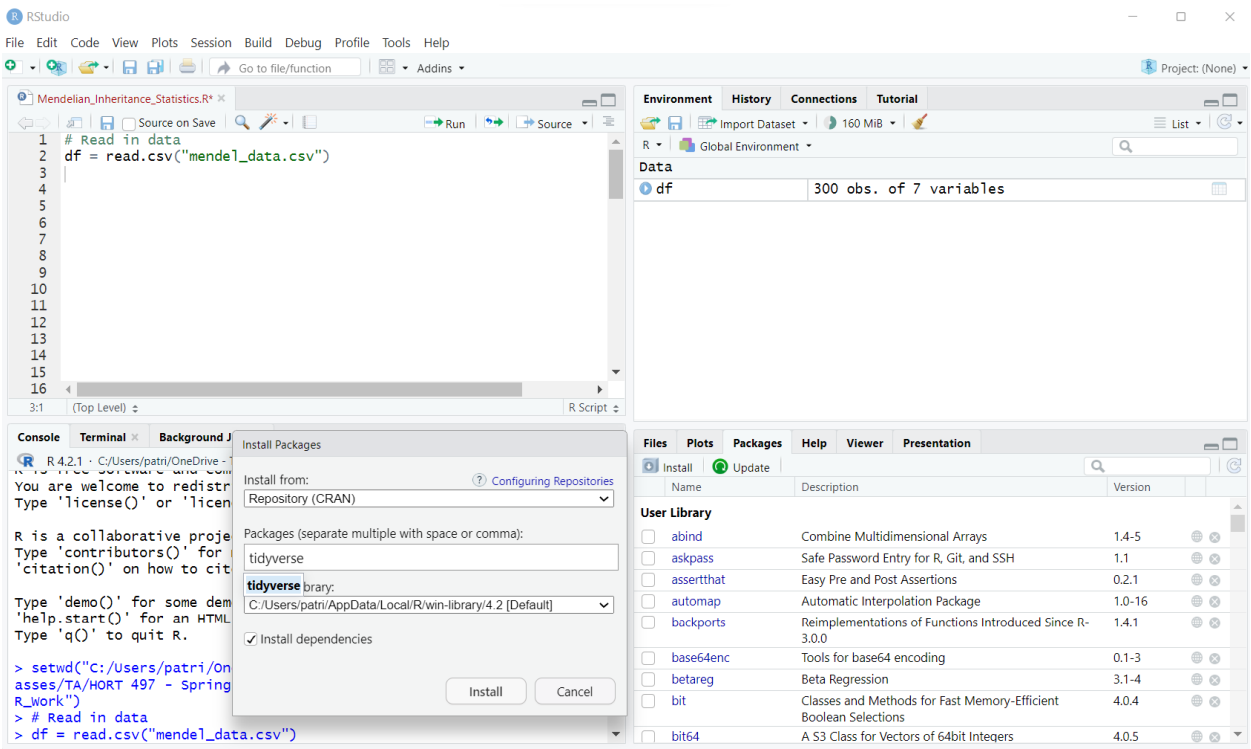
After downloading the mendel_data.xlsx data sheet from Canvas, open the file to view both sheets. Notice how the 'datasheet contains replicated data from Mendel's experiments while the 'metadata' sheet contains descriptions of the crosses that produced these plants...

The preferred data format to read into R is comma-separated value (.csv) files. To convert the plant data from .xlsx to .csv simply save the 'data' sheet as a .csv file using Excel or Google Sheets and place the file into your working directory on your computer. Once the .csv file is placed in your working directory, you can read the data into R using the read.csv() function by calling the file. After running your code line by line with the Run button or all at once by highlighting your code and using Ctrl+Enter, your data will be assigned to the object called 'df' for 'data frame'. Note how the text following the # did not get run in the Console... Using the # allows us to annotate our code and describe what it does. Also, notice how the df object appears in the Environment panel. You can click the df object in the Environment panel to view the data frame within RStudio to ensure it was read properly.

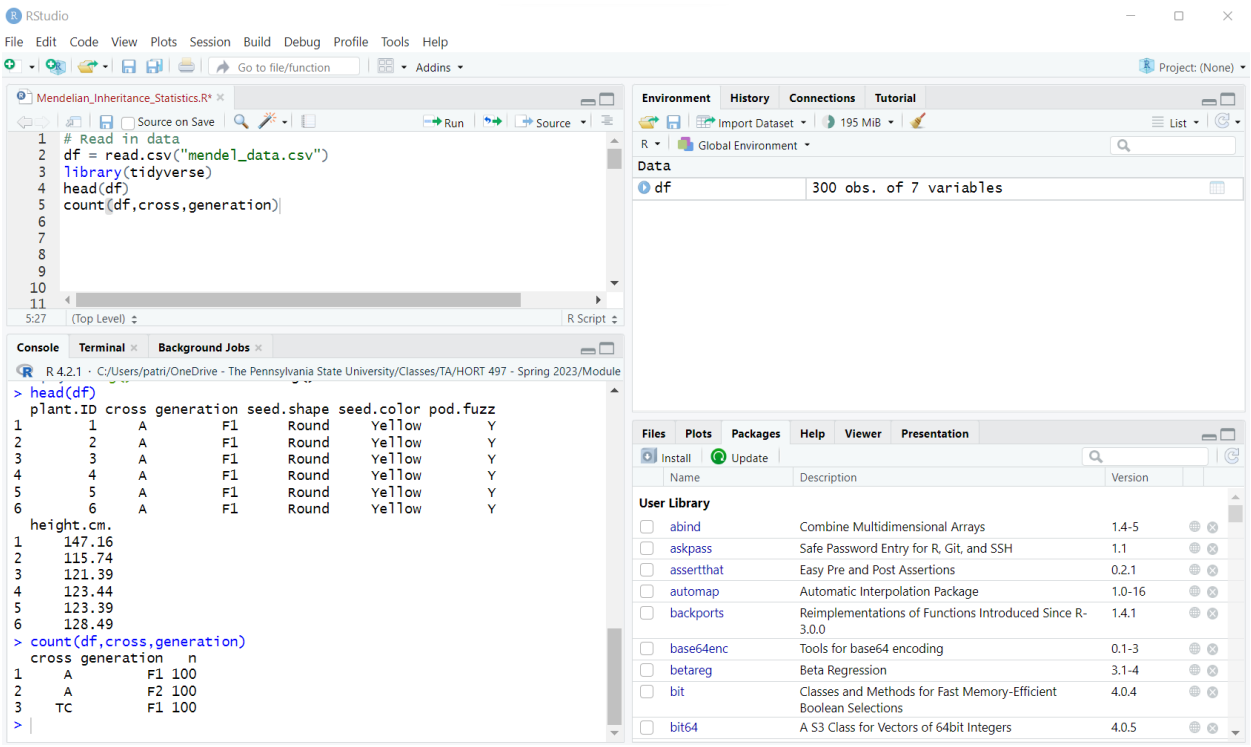


Analyzing Plant Phenotypes and Testing Mendel's Laws in R

After we have loaded Mendel's data into R, we can begin to take a look at what was recorded and analyze it with modern statistical software techniques! To begin, we need to download some additional components called Packages. Packages are like add-ons or extensions with many functions that allow you to do many more things with less code in R. For this activity, we will download a collection of packages called the tidyverse that aid data scientists with many shortcuts. Installing packages can be done within RStudio by clicking the Install button in the Packages tab on the bottom right panel and searching for the package you wish to download.

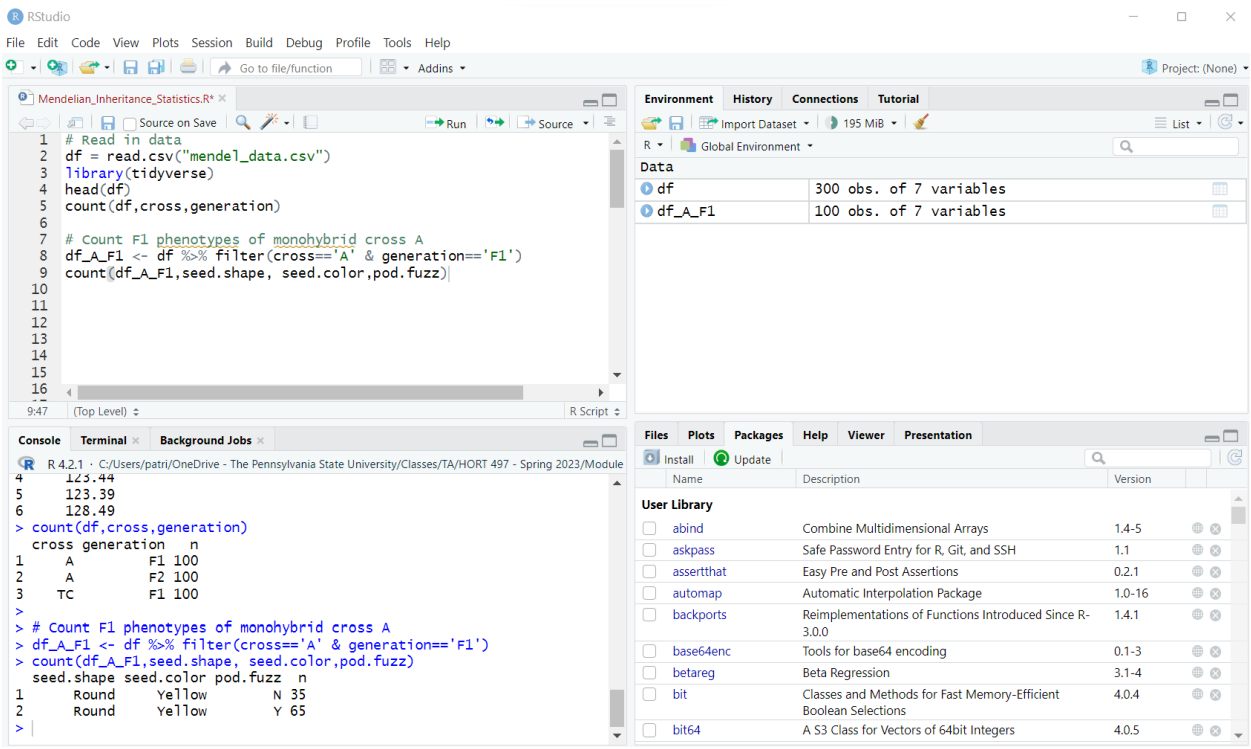


Once we have successfully downloaded the tidyverse packages, we can begin taking a look at Mendel's data with the head() and count() functions. First, we must call the package in our code with library(tidyverse). The head() function displays the first few lines of the data frame, and the count() function allows us to count unique values of one or more variables in a dataset. The first term in the count() function identifies the data frame that will be counted and the following terms, separated by commas, identify the columns in which unique values will be counted.



After running this code, we can see there are four phenotypes present in the data: seed shape, seed color, pod fuzz, and height reported in cm. We can also see there are three sets of plant data. Recall that cross A is a hybrid cross with one parent exhibiting the Round/Yellow/Fuzzy Pod/Tall phenotype and the other a Wrinkled/Green/Smooth Pod/Dwarf phenotype. The TC cross represents the cross between an F1 parent and an F2 parent with a Wrinkled/Green/Smooth Pod/Dwarf phenotype.

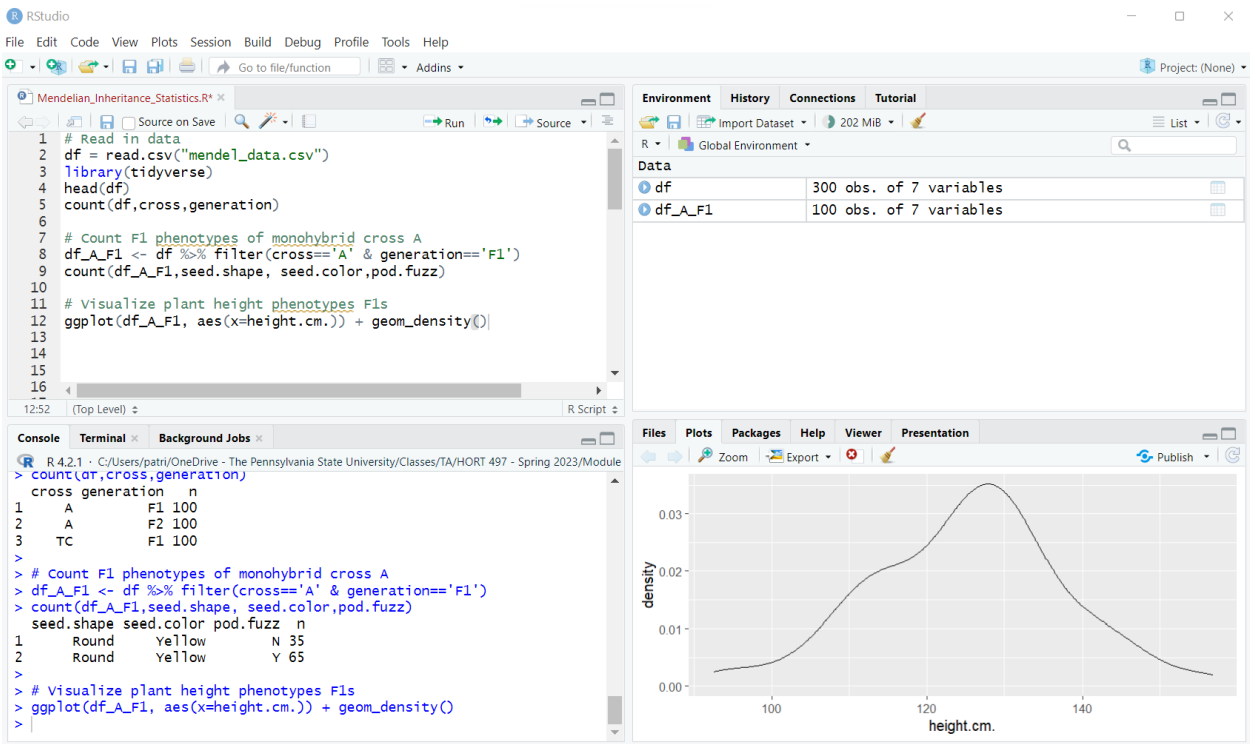
We can create a new dataset containing only F1s from the A cross by filtering the entire data set. We can do this by "piping" the original df into the filter() function using a pipe (%>%). Then, the filter function acts to select rows within the data frame where cross = "A" AND the generation = "F1". After storing this filtered dataset in df_A_F1, we can use the count() function once again to count the number of individuals with each combination of phenotypes.



Do all traits appear to be following the laws of Mendelian inheritance in these F1 offspring? Write this question and your answer in your .R script by annotating with a #.

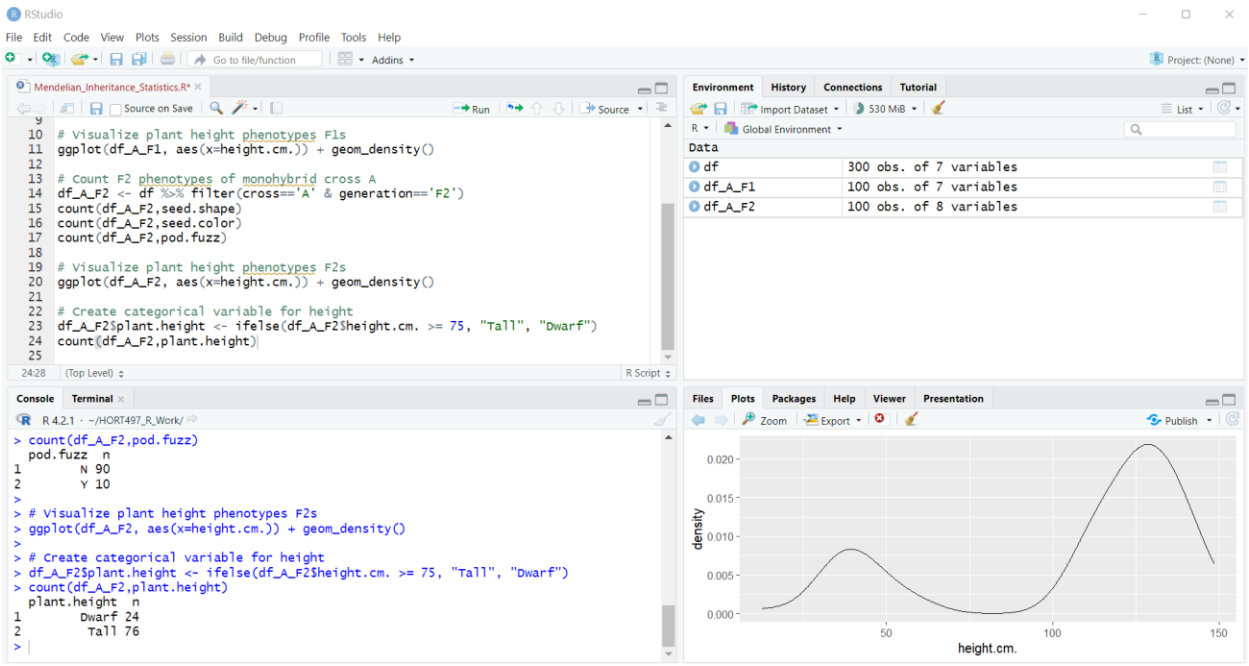
We can also visualize the distribution of the continuous variables plant height using ggplot: a data visualization package within the tidyverse. The ggplot() function requires us to note a data frame (df_A_F1) and we must at least note an x or y variables with the aes() aesthetics argument. To create a density plot, we must add an additional 'element' to our ggplot object: geom_density(). Also try using geom_histogram().

Does plant height appear to be following the laws of Mendelian inheritance in these F1 offspring? Write this question and your answer in your .R script by annotating with a #.



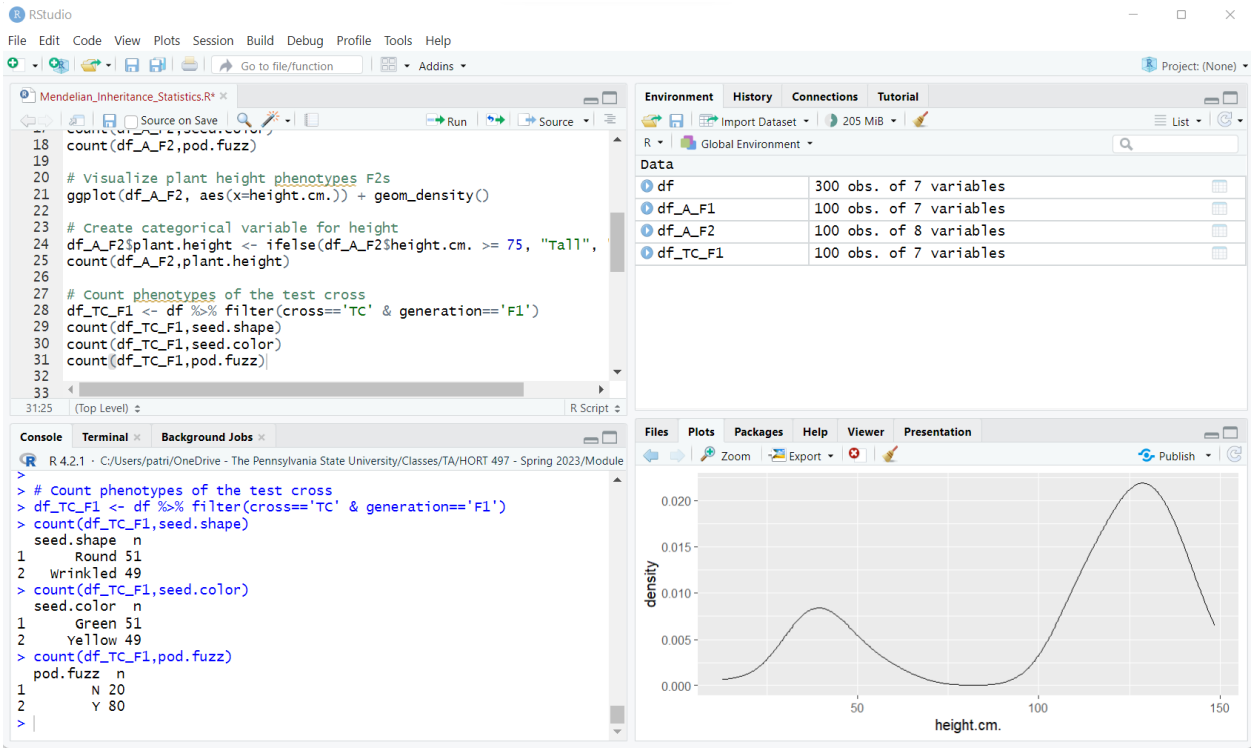
Next we can filter the original data frame once again to analyze the F2 generation. This time we can notice a clear presence of dwarf plants in the F2 offspring. But how many are there? To investigate this, we can create a categorical variable with an if else statement using the ifelse() function. The function works by creating one output ("Tall") if a test is passed (if height.cm. is greater or equal to 75) and another output ("Dwarf") if the test is not passed. That is, if height.cm. is less than 75. This output is then stored in a new column in our dataset called plant.height (df_A_F2$plant.height).

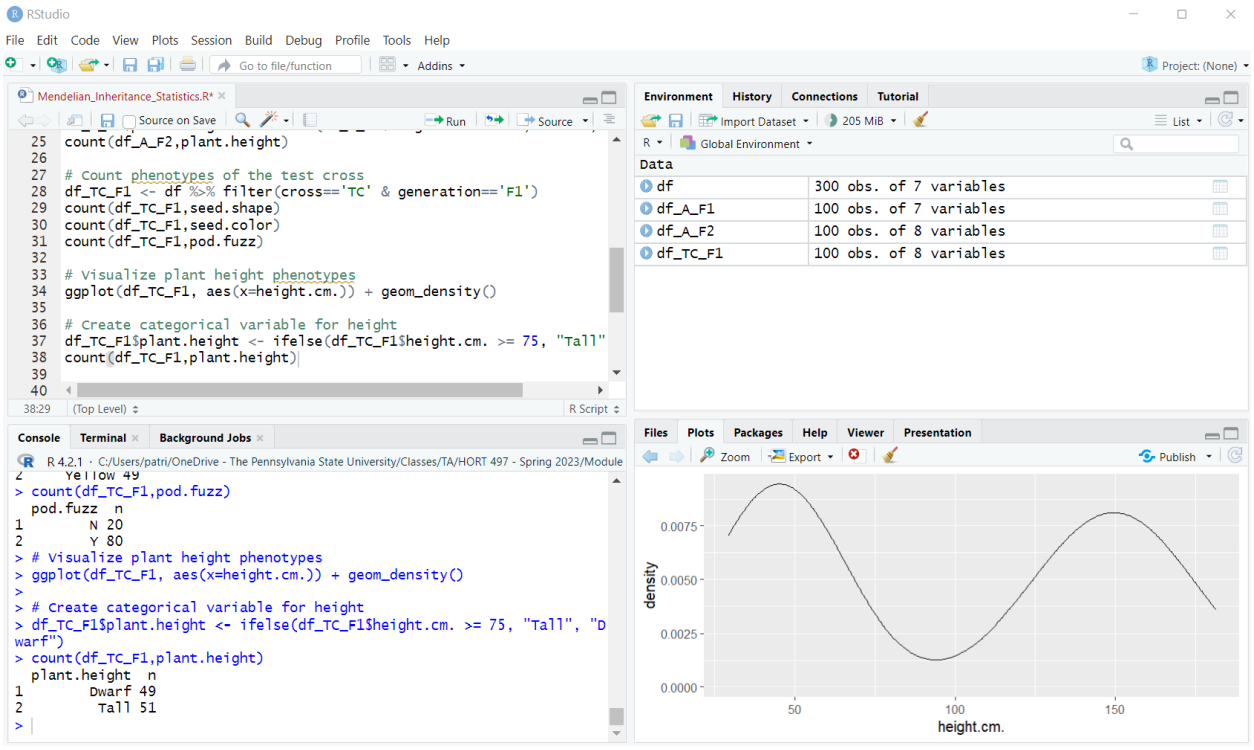
Do all traits appear to be following the laws of Mendelian inheritance in these F2 offspring? Write this question and your answer in your .R script by annotating with a #.



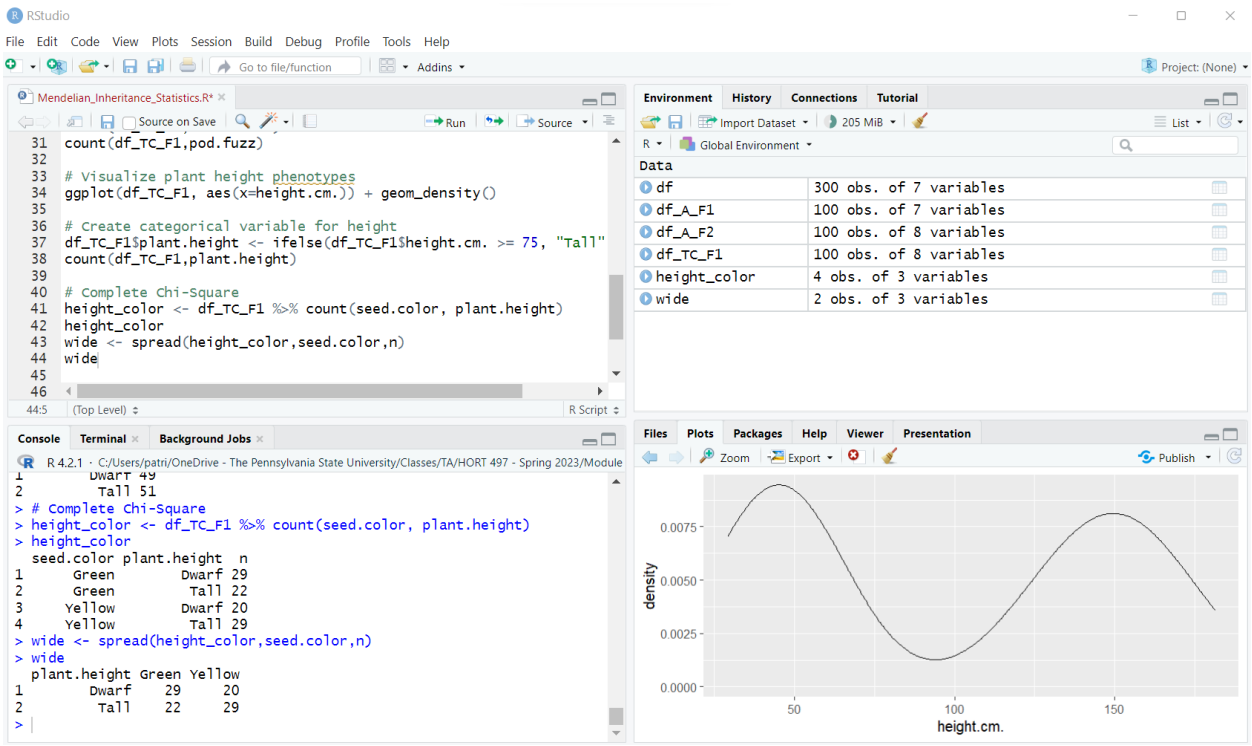
A "test cross" of the hybrid cross allows us to test Mendel's second law of independent assortment. Crossing an F1 of a hybrid cross with an individual with the recessive phenotype of the F2 generation should yield progeny with all possible combinations of dominant and recessive phenotypes in equal proportions. Furthermore, the gene for each trait should be sorted independently of genes for other traits meaning that the inheritance of one phenotypic trait is independent of the inheritance of another phenotypic trait.

In R, we can begin an analysis of the test cross by counting the occurrence of each individual phenotypic trait and determining if they occur in a 1:1 ratio.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Mendelian_Inheritance_Statistics.R*
count(df_A_F2,pod.fuzz)
# Visualize plant height phenotypes F2s
ggplot(df_A_F2, aes(x=height.cm.)) + geom_density()
# Create categorical variable for height
df_A_F2$plant.height <- ifelse(df_A_F2$height.cm. >= 75, "Tall",
count(df_A_F2,plant.height)
# Count phenotypes of the test cross
df_TC_F1 <- df %>% filter(cross=='TC' & generation=='F1')
count(df_TC_F1,seed.shape)
count(df_TC_F1,seed.color)
count(df_TC_F1,pod.fuzz)
Environment History Connections Tutorial
Data
df 300 obs. of 7 variables
df_A_F1 100 obs. of 7 variables
df_A_F2 100 obs. of 8 variables
df_TC_F1 100 obs. of 7 variables
Console Terminal Background Jobs
> # Count phenotypes of the test cross
> df_TC_F1 <- df %>% filter(cross=='TC' & generation=='F1')
> count(df_TC_F1,seed.shape)
seed.shape n
1 Round 51
2 Wrinkled 49
> count(df_TC_F1,seed.color)
seed.color n
1 Green 51
2 Yellow 49
> count(df_TC_F1,pod.fuzz)
pod.fuzz n
1 N 20
2 Y 80
Files Plots Packages Help Viewer Presentation
density
height.cm.

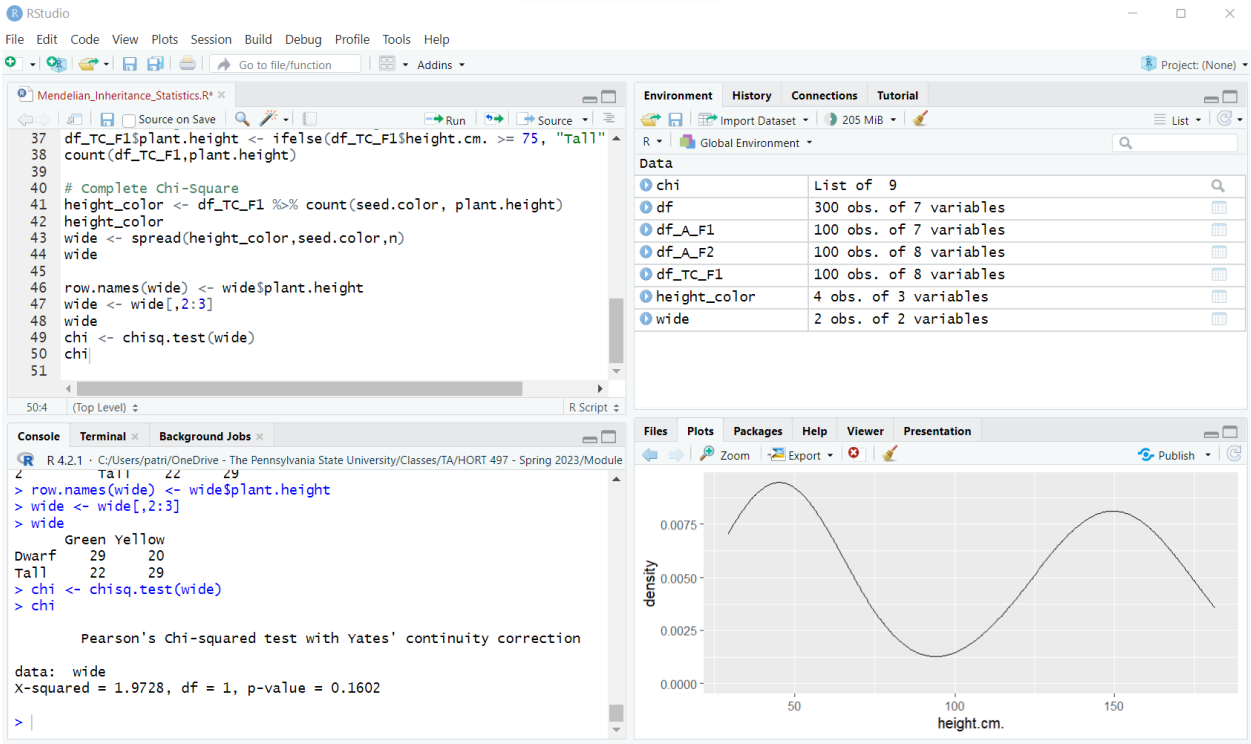

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Mendelian_Inheritance_Statistics.R*
count(df_A_F2,plant.height)
# Count phenotypes of the test cross
df_TC_F1 <- df %>% filter(cross=='TC' & generation=='F1')
count(df_TC_F1,seed.shape)
count(df_TC_F1,seed.color)
count(df_TC_F1,pod.fuzz)
# Visualize plant height phenotypes
ggplot(df_TC_F1, aes(x=height.cm.)) + geom_density()
# Create categorical variable for height
df_TC_F1$plant.height <- ifelse(df_TC_F1$height.cm. >= 75, "Tall"
count(df_TC_F1,plant.height)
Environment History Connections Tutorial
Data
df 300 obs. of 7 variables
df_A_F1 100 obs. of 7 variables
df_A_F2 100 obs. of 8 variables
df_TC_F1 100 obs. of 8 variables
Console Terminal Background Jobs
> count(df_TC_F1,pod.fuzz)
pod.fuzz n
1 N 20
2 Y 80
> # Visualize plant height phenotypes
> ggplot(df_TC_F1, aes(x=height.cm.)) + geom_density()
> # Create categorical variable for height
> df_TC_F1$plant.height <- ifelse(df_TC_F1$height.cm. >= 75, "Tall", "Dwarf")
> count(df_TC_F1,plant.height)
plant.height n
1 Dwarf 49
2 Tall 51
Files Plots Packages Help Viewer Presentation
density
height.cm.

A Chi-Square test for independence can be used to determine if there is any significant association between two categorical variables using contingency tables. These contingency tables note the number of occurrences of different combinations of two categorical variables. You can read more about Chi-Square tests in R here. In R, we can create contingency tables by manipulating the output of the count() function using the spread() function that is part of the dplyr package of tidyverse.



Then, we must manipulate this newly formatted data once again to obtain a data frame that only contains integers to utilize the chisq.test() function. To do this, we replace the numbered row names with the plant.height phenotypes on line 45 and select only the last two columns of the data frame on line 46. Values from data frames can be retrieved by referring to them in the following format df[rows,columns] where df[,2:3] refers to ALL rows of columns 2 through 3.

The results 'wide' data frame can be inputted right into the chisq.test() function, and we can store its output in the 'chi' variable. By simply running 'chi' to display this output, we can see the results of the chi-square test below...



Remember that a chi-square test for independence tests the null hypothesis that there is no association between the categorical variables. A p-value greater than 0.05 does not allow us to reject this null hypothesis leaving us to conclude that there is no association between the inheritance of the seed color and plant height trait.

Repeat this process for another combination of traits. Do the progeny of the test cross suggest these traits follow Mendel's laws?