Product\_Analysis\_for\_Chemistry

Emmanuel Duarte

2023-01-05

#install.packages("reshape2")  
library(reshape2)  
#install.packages("tidyverse")  
library("tidyverse")

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.5   
## ✔ 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()

tinytex::install\_tinytex(force = TRUE)  
chem <- read.csv("C:\\Users\\Owner\\OneDrive\\Desktop\\databasesforfun\\AnrescoSamples1021to0123.csv")  
#view(chem)  
  
colnames(chem)

## [1] "Sample.ID"   
## [2] "Description"   
## [3] "Lot.Batch.."   
## [4] "S.Code"   
## [5] "Category"   
## [6] "Sub.Category"   
## [7] "Strain.Name"   
## [8] "Strain.Type"   
## [9] "Batch.Size"   
## [10] "Sample.increments"   
## [11] "Weight..g."   
## [12] "Order.ID"   
## [13] "Sampled.Date"   
## [14] "Received.Date"   
## [15] "Completed.Date"   
## [16] "Sampling.Address"   
## [17] "Cultivation.Manufacturing.Address"   
## [18] "Collection.Type"   
## [19] "Shipping.Carrier"   
## [20] "Sample.Condition"   
## [21] "PO.."   
## [22] "Turnaround.Time"   
## [23] "Est..Completion.Date"   
## [24] "Special.Instructions"   
## [25] "Sample.Pass.Fail"   
## [26] "Cannabinoid.Profile.Pass.Fail"   
## [27] "Microbiological.Screen..Inhalables..Pass.Fail"   
## [28] "Microbiological.Screen.Non.Inhalables..Pass..Fail"  
## [29] "Pesticide.Screen.Phase.2.Pass..Fail"   
## [30] "Residual.Solvents.Phase.2.Pass..Fail"   
## [31] "Heavy.Metals.Pass.Fail"   
## [32] "Mycotoxins.Pass.Fail"   
## [33] "Water.Activity.Pass.Fail"   
## [34] "Foreign.Material.Pass.Fail"   
## [35] "Moisture.Pass.Fail"   
## [36] "Water.Activity"   
## [37] "Moisture.."   
## [38] "d8.THC..mg.g."   
## [39] "d8.THC..mg.ml."   
## [40] "d8.THC..dry..Percent"   
## [41] "d8.THC.mg.serving"   
## [42] "d8.THC.mg.package"   
## [43] "d9.THC..mg.g."   
## [44] "d9.THC..mg.ml."   
## [45] "d9.THC..dry..Percent"   
## [46] "d9.THC.mg.serving"   
## [47] "d9.THC.mg.package"   
## [48] "THCA..mg.g."   
## [49] "THCA..mg.ml."   
## [50] "THCA..dry..Percent"   
## [51] "THCA.mg.serving"   
## [52] "THCA.mg.package"   
## [53] "THCV..mg.g."   
## [54] "THCV..mg.ml."   
## [55] "THCV..dry..Precent"   
## [56] "THCV..mg.serving."   
## [57] "THCV..mg.package."   
## [58] "THCVA..mg.g."   
## [59] "THCVA..mg.ml."   
## [60] "THCVA..dry..Precent"   
## [61] "THCVA..mg.serving."   
## [62] "THCVA..mg.package."   
## [63] "CBD..mg.g."   
## [64] "CBD..mg.ml."   
## [65] "CBD..dry..Precent"   
## [66] "CBD..mg.serving."   
## [67] "CBD..mg.package."   
## [68] "CBDA..mg.g."   
## [69] "CBDA..mg.ml."   
## [70] "CBDA..dry..Precent"   
## [71] "CBDA..mg.serving."   
## [72] "CBDA..mg.package."   
## [73] "CBDV..mg.g."   
## [74] "CBDV..mg.ml."   
## [75] "CBDV..dry..Precent"   
## [76] "CBDV..mg.serving."   
## [77] "CBDV..mg.package."   
## [78] "CBN..mg.g."   
## [79] "CBN..mg.ml."   
## [80] "CBN..dry..Precent"   
## [81] "CBN..mg.serving."   
## [82] "CBN..mg.package."   
## [83] "CBG..mg.g."   
## [84] "CBG..mg.ml."   
## [85] "CBG..dry..Precent"   
## [86] "CBG..mg.serving."   
## [87] "CBG..mg.package."   
## [88] "CBC..mg.g."   
## [89] "CBC..mg.ml."   
## [90] "CBC..dry..Precent"   
## [91] "CBC..mg.serving."   
## [92] "CBC..mg.package."   
## [93] "CBCA..mg.g."   
## [94] "CBCA..mg.ml."   
## [95] "CBCA..dry..Precent"   
## [96] "CBCA..mg.serving."   
## [97] "CBCA..mg.package."   
## [98] "CBGA..mg.g."   
## [99] "CBGA..mg.ml."   
## [100] "CBGA..dry..Precent"   
## [101] "CBGA..mg.serving."   
## [102] "CBGA..mg.package."   
## [103] "Total.THC.mg.g"   
## [104] "Total.THC.mg.ml"   
## [105] "Total.THC..dry..mg.g"   
## [106] "Total.THC..dry..mg.ml"   
## [107] "Total.THC.Percent"   
## [108] "Total.THC..dry..Percent"   
## [109] "Total.THC..mg.serving."   
## [110] "Total.THC..mg.package."   
## [111] "Total.CBD.mg.g"   
## [112] "Total.CBD.mg.ml"   
## [113] "Total.CBD..dry..mg.g"   
## [114] "Total.CBD..dry..mg.ml"   
## [115] "Total.CBD.Percent"   
## [116] "Total.CBD..dry..Percent"   
## [117] "Total.CBD..mg.serving."   
## [118] "Total.CBD..mg.package."   
## [119] "Total.mg.g"   
## [120] "Total.mg.ml"   
## [121] "Total..dry..mg.g"   
## [122] "Total..dry..mg.ml"   
## [123] "Total.Percent"   
## [124] "Total..dry..Percent"   
## [125] "Total..mg.serving."   
## [126] "Total..mg.package."   
## [127] "Terpinolene...."   
## [128] "Eudesmol...."   
## [129] "α.Pinene...."   
## [130] "β.Pinene...."   
## [131] "Camphene...."   
## [132] "Eucalyptol...."   
## [133] "Isopulegol...."   
## [134] "Menthol...."   
## [135] "Borneol...."   
## [136] "Trans.Beta.Ocimene...."   
## [137] "Cis.Beta.Ocimene...."   
## [138] "Geraniol...."   
## [139] "Linalool...."   
## [140] "Guaiol...."   
## [141] "α.Bisabolol...."   
## [142] "α.Terpinene...."   
## [143] "Citronellol...."   
## [144] "δ.3.Carene...."   
## [145] "Cis.Nerolidol...."   
## [146] "Trans.Nerolidol...."   
## [147] "Limonene...."   
## [148] "P.Cymene...."   
## [149] "γ.Terpinene...."   
## [150] "α.Terpineol...."   
## [151] "ß.Myrcene...."   
## [152] "CaryophylleneOxide...."   
## [153] "α.Humulene...."   
## [154] "ß.Caryophyllene...."   
## [155] "Aerobic.Plate.Count..cfu.g."   
## [156] "Yeast..cfu.g."   
## [157] "Mold..cfu.g."   
## [158] "Coliforms..cfu.g."   
## [159] "E..Coli..cfu.g."   
## [160] "Salmonella..per.1g."   
## [161] "Listeria..per.1g."   
## [162] "Pseudomonas..cfu.g."   
## [163] "Aspergillus"   
## [164] "Aspergillus.Fumigatus"   
## [165] "Aspergillus.Flavus"   
## [166] "Aspergillus.Niger"   
## [167] "Aspergillus.Terreus"   
## [168] "X2.Phenylphenol"   
## [169] "X3.Hydroxycarbofuran"   
## [170] "X4.4..Methoxychlor.olefin"   
## [171] "X5.OH.Thiabendazole."   
## [172] "Abamectin"   
## [173] "Acephate"   
## [174] "Acequinocyl."   
## [175] "Acetamiprid"   
## [176] "Acetochlor"   
## [177] "Alachlor"   
## [178] "Aldicarb"   
## [179] "Aldicarb.Sulfone"   
## [180] "Aldicarb.sulfoxide."   
## [181] "Aldrin."   
## [182] "Atrazine"   
## [183] "Azoxystrobin"   
## [184] "BHC.alpha"   
## [185] "BHC.beta"   
## [186] "Benfluralin"   
## [187] "Bifenazate"   
## [188] "Bifenthrin"   
## [189] "Bitertanol"   
## [190] "Boscalid."   
## [191] "Buprofezin"   
## [192] "Captan"   
## [193] "Carbaryl"   
## [194] "Carbendazim..MBC."   
## [195] "Carbofuran"   
## [196] "Carfentrazone.ethyl"   
## [197] "Chlorantraniliprole"   
## [198] "Chlordane.cis"   
## [199] "Chlordane.trans"   
## [200] "Chlorfenapyr"   
## [201] "Chlorobenzilate"   
## [202] "Chlorotoluron"   
## [203] "Chlorpropham"   
## [204] "Chlorpyrifos"   
## [205] "Chlorpyrifos.methyl"   
## [206] "Chlorthiophos"   
## [207] "Clethodim"   
## [208] "Clofentezine"   
## [209] "Clomazone"   
## [210] "Clothianidin"   
## [211] "Coumaphos"   
## [212] "Cyazofamid."   
## [213] "Cyfluthrin"   
## [214] "Cymoxanil."   
## [215] "Cypermethrin"   
## [216] "Cyprodinil."   
## [217] "Cyromazine"   
## [218] "DCPA..Dacthal."   
## [219] "Daminozide.."   
## [220] "Diallate"   
## [221] "Diazinon"   
## [222] "Dichlorvos"   
## [223] "Dieldrin"   
## [224] "Diflubenzuron"   
## [225] "Dimethachlor"   
## [226] "Dimethoate."   
## [227] "Dimethomorph"   
## [228] "Diniconazole"   
## [229] "Dinotefuran"   
## [230] "Diphenamid"   
## [231] "Diphenylamine..DPA."   
## [232] "Diuron"   
## [233] "Esfenvalerate"   
## [234] "Ethalfluralin"   
## [235] "Ethoprophos"   
## [236] "Ethylan"   
## [237] "Etofenprox"   
## [238] "Etoxazole"   
## [239] "Etridazole."   
## [240] "Fenarimol"   
## [241] "Fenbuconazole"   
## [242] "Fenhexamid"   
## [243] "Fenoxycarb"   
## [244] "Fenpropimorph"   
## [245] "Fenpyroximate."   
## [246] "Fenthion"   
## [247] "Fenvalerate"   
## [248] "Fipronil"   
## [249] "Flonicamid."   
## [250] "Fluazifop.P.butyl"   
## [251] "Flubendiamide"   
## [252] "Fluchloralin"   
## [253] "Flucythrinate"   
## [254] "Fludioxanil"   
## [255] "Fluoxastrobin"   
## [256] "Flusilazole"   
## [257] "Flutolanil"   
## [258] "Flutriafol"   
## [259] "Fonofos."   
## [260] "Formetanate.Hydrochloride"   
## [261] "Heptachlor"   
## [262] "Hexachlorobenzene"   
## [263] "Hexaconazole"   
## [264] "Hexazinone"   
## [265] "Hexythiazox..a.."   
## [266] "Imazalil"   
## [267] "Imidacloprid"   
## [268] "Indoxacarb."   
## [269] "Isazophos"   
## [270] "Isopropalin"   
## [271] "Isoproturon"   
## [272] "Kresoxim.Methyl"   
## [273] "Lindane"   
## [274] "Linuron."   
## [275] "MGK.264."   
## [276] "Malathion"   
## [277] "Mandipropamid"   
## [278] "Metalaxyl"   
## [279] "Methamidophos"   
## [280] "Methiocarb"   
## [281] "Methomyl"   
## [282] "Methoxychlor."   
## [283] "Methoxyfenozide"   
## [284] "Metolachlor"   
## [285] "Mevinphos"   
## [286] "Mirex."   
## [287] "Myclobutanil."   
## [288] "Naled..b..."   
## [289] "Novaluron"   
## [290] "Omethoate."   
## [291] "Oxadiazon"   
## [292] "Oxamyl"   
## [293] "Oxyflourfen"   
## [294] "Paclobutrazol"   
## [295] "Parathion"   
## [296] "Parathion.Methyl."   
## [297] "Pebulate."   
## [298] "Penconazole"   
## [299] "Pendimethalin"   
## [300] "Pentachloroaniline"   
## [301] "Pentachloroanisole"   
## [302] "Pentachlorobenzene"   
## [303] "Pentachlorobenzonitrile"   
## [304] "Pentachloronitrobenzene"   
## [305] "Permethrins"   
## [306] "Phosmet"   
## [307] "Piperonyl.Butoxide"   
## [308] "Pirimicarb"   
## [309] "Pirimiphos.methyl"   
## [310] "Pirimiphos.ethyl"   
## [311] "Prallethrin.."   
## [312] "Pretilachlor"   
## [313] "Prochloraz."   
## [314] "Procymidone"   
## [315] "Propamocarb"   
## [316] "Propargite"   
## [317] "Propiconazole"   
## [318] "Propisochlor"   
## [319] "Propoxur"   
## [320] "Propyzamide"   
## [321] "Prothiofos"   
## [322] "Pymetrozine"   
## [323] "Pyraclostrobin"   
## [324] "Pyretrhins"   
## [325] "Pyridaben"   
## [326] "Pyrimethanil"   
## [327] "Pyriproxyfen"   
## [328] "Quinalphos"   
## [329] "Quintozene"   
## [330] "Spinetoram"   
## [331] "Spinosad"   
## [332] "Spiromesifen"   
## [333] "Spirotetramat"   
## [334] "Spiroxamine"   
## [335] "Sulfentrazone"   
## [336] "Sulfotep"   
## [337] "Tau.Fluvalinate"   
## [338] "Tebuconazole"   
## [339] "Tebufenozide"   
## [340] "Tecnazene"   
## [341] "Tefluthrin"   
## [342] "Terbutylazine"   
## [343] "Tetrachloroaniline..2.3.5.6."   
## [344] "Tetradifon"   
## [345] "Thiaclorprid"   
## [346] "Thiamethoxam"   
## [347] "Thiobencarb"   
## [348] "Tolclofos.methyl"   
## [349] "Transfluthrin"   
## [350] "Triadimefon"   
## [351] "Triallate"   
## [352] "Trichlorfon"   
## [353] "Tricyclazole"   
## [354] "Trifloxystrobin"   
## [355] "Triflumizole"   
## [356] "Trifluralin"   
## [357] "Vamidothion"   
## [358] "Vinclozolin"   
## [359] "Isobutane..ppm."   
## [360] "n.Butane..ppm."   
## [361] "n.Pentane..ppm."   
## [362] "Ethanol..ppm."   
## [363] "Acetone..ppm."   
## [364] "Acetonitrile..ppm."   
## [365] "n.Hexane..ppm."   
## [366] "Tetrahydrofuran..ppm."   
## [367] "Chloroform..ppm."   
## [368] "Carbon.Tetrachloride..ppm."   
## [369] "Benzene..ppm."   
## [370] "n.Heptane..ppm."   
## [371] "Toluene..ppm."   
## [372] "M.P.Xylene..ppm."   
## [373] "O.Xylene..ppm."   
## [374] "Methanol..ppm."   
## [375] "X1.2.Dichloroethane..ppm."   
## [376] "Ethyl.Acetate..ppm."   
## [377] "Ethyl.Ether..ppm."   
## [378] "Ethylene.Oxide..ppm."   
## [379] "Isopropyl.Alcohol..ppm."   
## [380] "Methylene.Chloride..ppm."   
## [381] "Propane..ppm."   
## [382] "Trichloroethylene..ppm."   
## [383] "Total.Xylenes"   
## [384] "Arsenic..ppm."   
## [385] "Arsenic.Pass.Fail"   
## [386] "Cadmium..ppm."   
## [387] "Cadmium.Pass.Fail"   
## [388] "Lead..ppm."   
## [389] "Lead.Pass.Fail"   
## [390] "Mercury..ppm."   
## [391] "Mercury.Pass.Fail"   
## [392] "Total.Aflatoxins"   
## [393] "Total.Aflatoxins.Pass.Fail"   
## [394] "Ochratoxins"   
## [395] "Ochratoxins.Pass.Fail"   
## [396] "Sand..Soils..Cinders..and.Dirt"   
## [397] "Sand..Soils..Cinders..and.Dirt.Pass.Fail"   
## [398] "Imbedded.foreign.materail"   
## [399] "Imbedded.foreign.materail.Pass.Fail"   
## [400] "Insect.fragment"   
## [401] "Insect.fragment.Pass.Fail"   
## [402] "Mammalian.excreta"   
## [403] "Mammalian.excreta.Pass.Fail"   
## [404] "Hair"   
## [405] "Hair.Pass.Fail"

#view(chem)  
getwd()

## [1] "C:/Users/Owner/OneDrive/Desktop/ProductAnalysisforChemistry"

As the previous lab director for “Chemistry” I have access to all the Quality Control(QC) analytical tests as well as the Certificate of Analysis (COA) test results for every product manufactured during the time the company has been active. Chemistry is a cannabis extraction company and during my time there I was in charge of the QC at every step of production, from intake to final product. These test results show the cannabinoid concentrations for Flower, Tinctures, Various Edibles, Vapor Cartridges(Live Resin and FSO(Full Spectrum Oil)), Cannabis Powder (for pressing pills), and Distillate (Isolated d9THC). As a part of this report I was contracted by the CEO and Marketing director to answer some interesting Marketing questions and see if there has been something hiding in this data that that could be used. Chemistry based their niche in the market on something called Full-Spectrum; the claim is that its oil is the best representation of the “flower” in a concentrated form. This is because of the solvents we use and the fact that we do a relatively simply refinement process that allows for the retention of various cannabinoids (active chemicals found in Cannabis and Hemp eg. THC, CBD, THCA, CBDA, CBG, THCV,CBN just to name a few) There are about 24 cannabinoids that analytical labs test for although there are people still discovering new compounds as well as synthesizing them in a lab setting. Cannabis/Hemp also contain Essecial oils that we refer to as Terpenes. Terpenes influence the medicinal effect and give the flower its aroma and when smoked its flavor. Cannabinoids and terpenes are going to be the main focus of this analysis. I’ve been given the task to compare the ratios of these two groups of compounds from their flower form to their concentrated form in order to prove exactly how “full spectrum” their products are. I’ll try my best to explain the analysis process as it pertains to this somewhat esoteric data.

These reports give a very detailed and somewhat excessive amount of information back. My first step will be cleaning up the data to keep only relevant information.

Important Information for categorizing data. “Description” “Lot.Batch..” “Category” “Sub.Category” “Strain.Name” “Strain.Type”

Cannabinoids “d9.THC..mg.g.” “CBD..mg.g.” “CBDA..mg.g.” “d8.THC..mg.g.” “CBDV..mg.g.” “CBN..mg.g.” “CBG..mg.g.” “CBC..mg.g.” “CBCA..mg.g.” “CBGA..mg.g.” “Total.THC.mg.g” “Total.CBD.mg.g”

Terpenes “Terpinolene….” “Eudesmol….”  
“α.Pinene….” “β.Pinene….”  
“Camphene….” “Eucalyptol….”  
“Isopulegol….” “Menthol….”  
“Borneol….” “Trans.Beta.Ocimene….”  
“Cis.Beta.Ocimene….” “Geraniol….”  
“Linalool….” “Guaiol….”  
“α.Bisabolol….” “α.Terpinene….”  
“Citronellol….” “δ.3.Carene….”  
“Cis.Nerolidol….” “Trans.Nerolidol….”  
“Limonene….” “P.Cymene….”  
“γ.Terpinene….” “α.Terpineol….”  
“ß.Myrcene….” “CaryophylleneOxide….”  
“α.Humulene….” “ß.Caryophyllene….”

I found all the columns that I needed for this part of the analysis. I'll make a new df with just these values to be able to parse through it easier.  
  
  
```r  
simpleChem <- chem %>% select(Description,Lot.Batch.., Category, Sub.Category, Strain.Name, Strain.Type, d9.THC..mg.g.,THCA..mg.g., d8.THC..mg.g., THCV..mg.g.,THCVA..mg.g., CBD..mg.g., CBDA..mg.g., CBDV..mg.g., CBN..mg.g., CBG..mg.g., CBC..mg.g., CBCA..mg.g., CBGA..mg.g., Total.THC.mg.g, Total.CBD.mg.g, Terpinolene....,Eudesmol....,α.Pinene....,β.Pinene...., Camphene...., Eucalyptol...., Isopulegol...., Menthol...., Borneol...., Trans.Beta.Ocimene....,Cis.Beta.Ocimene....,Geraniol....,Linalool....,Guaiol...., α.Bisabolol...., α.Terpinene....,Citronellol....,δ.3.Carene....,Cis.Nerolidol....,Trans.Nerolidol....,Limonene...., P.Cymene...., γ.Terpinene....,α.Terpineol...., ß.Myrcene...., CaryophylleneOxide....,α.Humulene...., ß.Caryophyllene....)  
  
view(simpleChem)  
#write.csv(simpleChem, file = "simpleChem.csv")

I would like to first simply compare the total cannabinoids, but no such group exists in this dataframe. I will make a new column that adds all cannabinoids in mg/g format.

#I am unable to add these together because there are non-numeric values in the DF rows "ND" for the cannabinoids that were not detected. I will change these to NA to be able to work with those values.   
  
cannabinoid\_columns <- c("d9.THC..mg.g.", "THCA..mg.g.", "THCV..mg.g.", "THCVA..mg.g.", "d8.THC..mg.g.", "CBD..mg.g.", "CBDA..mg.g.", "CBDV..mg.g.", "CBN..mg.g.", "CBG..mg.g.", "CBC..mg.g.", "CBCA..mg.g.", "CBGA..mg.g.")  
  
view(cannabinoid\_columns)  
terps\_together <- c("Terpinolene....", "Eudesmol....",   
"α.Pinene...." , "β.Pinene...." ,   
"Camphene....", "Eucalyptol....",   
"Isopulegol....", "Menthol....",   
"Borneol....", "Trans.Beta.Ocimene....",   
"Cis.Beta.Ocimene....", "Geraniol....",   
"Linalool....", "Guaiol....",   
"α.Bisabolol....", "α.Terpinene....",   
"Citronellol....", "δ.3.Carene....",   
"Cis.Nerolidol....", "Trans.Nerolidol....",   
"Limonene....", "P.Cymene....",   
"γ.Terpinene....", "α.Terpineol....",   
"ß.Myrcene....", "CaryophylleneOxide....",   
"α.Humulene....", "ß.Caryophyllene....")  
  
simpleChem <- simpleChem %>%  
 mutate\_at(vars(cannabinoid\_columns), as.numeric)

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.  
## ℹ Please use `all\_of()` or `any\_of()` instead.  
## # Was:  
## data %>% select(cannabinoid\_columns)  
##   
## # Now:  
## data %>% select(all\_of(cannabinoid\_columns))  
##   
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.

## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion  
  
## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion

simpleChem <- simpleChem %>%  
 mutate(total\_cannabinoidsMg.g = rowSums(select(., cannabinoid\_columns), na.rm = TRUE))  
  
simpleChem <- simpleChem %>%  
 mutate\_at(vars(terps\_together), as.numeric)

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.  
## ℹ Please use `all\_of()` or `any\_of()` instead.  
## # Was:  
## data %>% select(terps\_together)  
##   
## # Now:  
## data %>% select(all\_of(terps\_together))  
##   
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion  
## NAs introduced by coercion

simpleChem <- simpleChem %>%  
 mutate(total\_terpsMg.g = rowSums(select(., terps\_together), na.rm = TRUE))  
  
view(simpleChem)  
  
  
#Now I created a simplified total cannabinoids column in mg/g as well as a total terpene content column in mg/g. I want to now group by strain name in order to compare tests of the same strain name but in flower form vs. concentrate form.

simpleChem <- simpleChem %>%  
 group\_by(Description)  
simpleChem <- simpleChem %>%  
 arrange(Description)  
  
  
#The strains that I can do a comparison of ratios between cannabinoid concentrations and terpene content between flower and concentrate form are only the ones that we did a QC test for before it was processed into an oil. There are not too many examples so I will list them all to do an analysis of each individual one.   
  
#Cannabinoids Only- Gelonade, Lemon Mac, Orange Durban, Strawnana Frosting, Apricot Trainwreck,Jungle Pound Cake  
  
#Cannabinoids and Terps- Lady Benbow, Pineapple Wonder, Purple Wreck, Royal Cherry Kush, Strawberry Banana Live, Super Lemon Haze Live, XJ13, ACDC Flower/ACDC Orally Consumed Concentrate (Same Flower infused into Coconut oil),Gas and Roses, Grease Monkey, Jack Herrer,   
  
#Terps only- Sapphire Tsu  
  
#Fresh vs Cured Terps-Gas and Roses,   
  
#Started doing the above and realized many of the columns don't have the strain name. I will Have to use the description to group and find matching pairs, but most of the rows in description have "Chemistry. " before the description. I will remove Chemistry to group them properly.   
  
simpleChem <- simpleChem %>%  
 mutate(Description = gsub("Chemistry. ", "", Description))  
  
simpleChem <- simpleChem %>%  
 mutate(Description = gsub("Chemistry ", "", Description))  
  
simpleChem <- simpleChem %>%  
 mutate(Description = gsub("Moon Made Flower - ", "", Description))  
  
view(simpleChem)  
#write.csv(simpleChem, file = "simplerChem.csv")

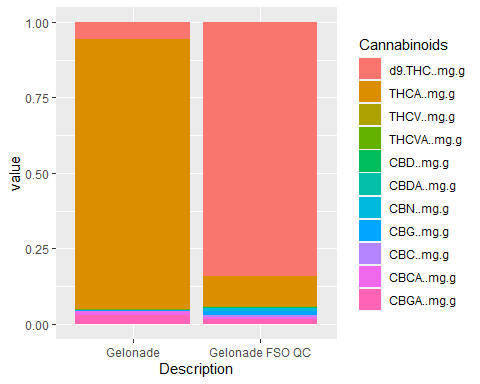
#I will now start working strain by strain comparing the ratios of total cannabinoids and total terpenes to see how similar they are from flower to concentrated form. I will start with only the strains that have data for both.

#Cannabinoid Only comparison between flower and fso.   
 Gelonade <-   
 simpleChem %>%  
 filter(grepl("Gelonade", Description)) %>%  
 group\_by(Description)  
#Removed any of the rows that did not have the data that I needed. I will continue to do so throughout the analysis. My Objective is to find rows that are the same strain and are flower as well as a finished packaged product. Usually Indicated by the term Cartridge or the size of the package.   
Gelonade <- Gelonade %>%  
 filter(Description != "Gelonade Flower QC")  
  
#Gelonade Individual Cannabinoid Formula for Columns 13 - 19  
#It made more sense to use the names of the columns rather than the column indexes.   
#Formula to find the proportions of cannabinoids to total cannabinoids for both the Gelonade flower and the Gelonade FSO QC.  
GelIndCannProp <- Gelonade%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
view(GelIndCannProp)  
#I am making another dataframe that contains only the second row from the GelIndCann dataframe with just the individual cannabinoid proportions for Gelonade.   
  
#Formula to select only the cannabinoid columns from the GelIndCannProp Dataframe.   
IndCannPropGelonade <- GelIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

#Substituted .\_IndCann for an empty string to make it look cleaner on the graph.   
colnames(IndCannPropGelonade) <- gsub(".\_IndCann", "", colnames(IndCannPropGelonade))  
  
  
#Needed to transpose this df to have it work in a stacked graph  
transposed\_IndCannPropGelonade <- t(IndCannPropGelonade)  
#view(transposed\_IndCannPropGelonade)  
  
#Needed to use the Melt function in the reshape2 package to change the dataframe to be able to make a stacked barchart.   
transposed\_IndCannPropGelonade\_melt <- melt(IndCannPropGelonade, id.vars = "Description")  
#view(transposed\_IndCannPropGelonade\_melt)  
#I changed the variable name to Cannabinoids to make it more clear on the dataframe.   
colnames(transposed\_IndCannPropGelonade\_melt)[colnames(transposed\_IndCannPropGelonade\_melt) == "variable"] <- "Cannabinoids"  
  
  
ggplot(data = transposed\_IndCannPropGelonade\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 9 rows containing missing values (`position\_stack()`).

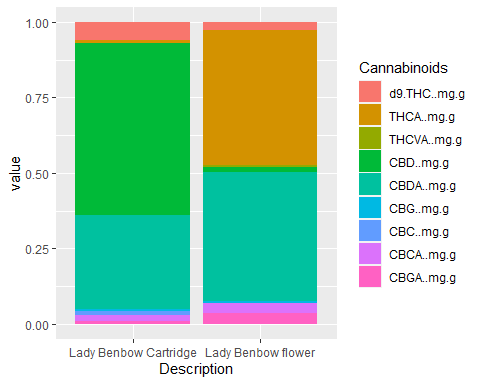


#Cannaninoid comparison between Flower and FSO for the Lady Benbow Strain.   
 LadyBenbow <-   
 simpleChem %>%  
 filter(grepl("Lady Benbow", Description)) %>%  
 group\_by(Description)  
  
  
#Filtered Out other Lady Benbows irrelevant to this study.   
LadyBenbow <- LadyBenbow %>%  
 filter(Description != "Lady Benbow FSO QC",  
 Description != "Lady Benbow Sleepy Tincture",  
 Description != "Lady Benbow Tincture",   
 Description != "Lady Benbow Tincture (30ml)",  
 total\_terpsMg.g != 0.00000  
 )  
#LadyBenbow  
#Lady Benbow Cannabinoid Proportion formulas for a comparison stacked bar chart.   
LadyIndCannProp <- LadyBenbow%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
view(LadyIndCannProp)  
  
IndCannPropLady <- LadyIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropLady) <- gsub(".\_IndCann", "", colnames(IndCannPropLady))  
  
  
  
transposed\_IndCannPropLady <- t(IndCannPropLady)  
#view(transposed\_IndCannPropLady)  
  
transposed\_IndCannPropLady\_melt <- melt(IndCannPropLady, id.vars = "Description")  
  
#Change the column Variable to Cannabinoids.   
colnames(transposed\_IndCannPropLady\_melt)[colnames(transposed\_IndCannPropLady\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropLady\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 10 rows containing missing values (`position\_stack()`).

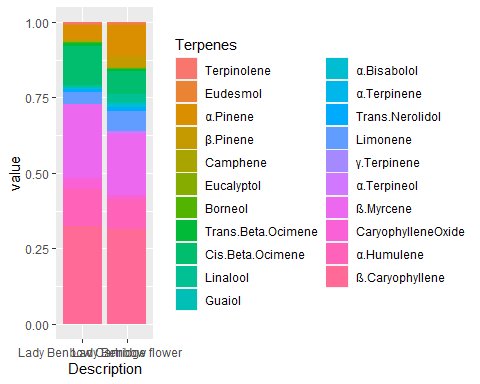


#Terpene comparison between Flower and FSO for the Lady Benbow Strain.   
 LadyBenbow <-   
 simpleChem %>%  
 filter(grepl("Lady Benbow", Description)) %>%  
 group\_by(Description)  
  
  
#Filtered Out other Lady Benbows irrelevant to this study.   
LadyBenbow <- LadyBenbow %>%  
 filter(Description != "Lady Benbow FSO QC",  
 Description != "Lady Benbow Sleepy Tincture",  
 Description != "Lady Benbow Tincture",   
 Description != "Lady Benbow Tincture (30ml)",  
 total\_terpsMg.g != 0.00000  
 )  
#view(LadyBenbow)  
  
#Lady Benbow Cannabinoid Proportion formulas for a comparison stacked bar chart.   
LadyIndTerpProp <- LadyBenbow%>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
#view(LadyIndTerpProp)  
  
IndTerpPropLady <- LadyIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

colnames(IndTerpPropLady) <- gsub("....\_IndTerp", "", colnames(IndTerpPropLady))  
  
#view(IndTerpPropLady)  
  
transposed\_IndTerpPropLady <- t(IndTerpPropLady)  
  
transposed\_IndTerpPropLady\_melt <- melt(IndTerpPropLady, id.vars = "Description")  
  
#Change the column Variable to Terpenes.   
colnames(transposed\_IndTerpPropLady\_melt)[colnames(transposed\_IndTerpPropLady\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropLady\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 17 rows containing missing values (`position\_stack()`).

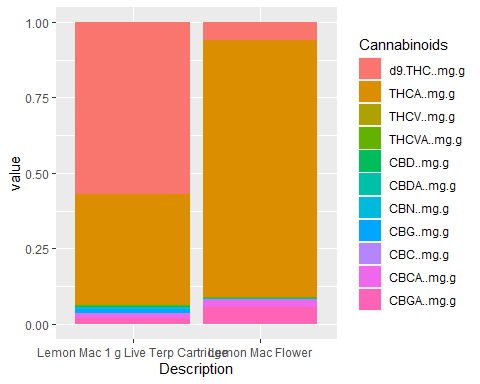


#Cannabinoid and terpene comparison between Flower and FSO.  
 LemonMac <-   
 simpleChem %>%  
 filter(grepl("Lemon Mac", Description)) %>%  
 group\_by(Description)  
  
LemonMac <- LemonMac %>%  
 filter(Description != "Lemon Mac FSO",  
 Description != "Lemon Mac FSO part2 QC",)  
  
#view(LemonMac)  
#Lemon Mac Cannabinoid Proportion formulas for a comparison stacked bar chart.   
LemonMIndCannProp <- LemonMac%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
#view(LemonMIndCannProp)  
  
IndCannPropLemonM <- LemonMIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

view(IndCannPropLemonM)  
  
colnames(IndCannPropLemonM) <- gsub(".\_IndCann", "", colnames(IndCannPropLemonM))  
  
  
transposed\_IndCannPropLemonM <- t(IndCannPropLemonM)  
#view(transposed\_IndCannPropLemonM)  
  
transposed\_IndCannPropLemonM\_melt <- melt(IndCannPropLemonM, id.vars = "Description")  
  
#Change the column Variable to Cannabinoids.   
colnames(transposed\_IndCannPropLemonM\_melt)[colnames(transposed\_IndCannPropLemonM\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropLemonM\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 4 rows containing missing values (`position\_stack()`).

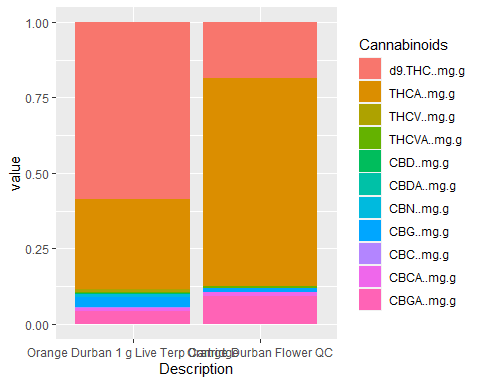


ODB <-   
 simpleChem %>%  
 filter(grepl("Orange Durban", Description)) %>%  
 group\_by(Description)  
view(ODB)  
  
ODB <- ODB %>%  
 filter(Description != "Orange Durban FSO",  
 Description != "Orange Durban FSO QC")  
  
  
#Orange Durban Cannabinoid Proportion formulas for a comparison stacked bar chart.   
ODBIndCannProp <- ODB%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
#view(ODBIndCannProp)  
  
IndCannPropODB <- ODBIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

#view(IndCannPropODB)  
  
colnames(IndCannPropODB) <- gsub(".\_IndCann", "", colnames(IndCannPropODB))  
  
  
transposed\_IndCannPropODB <- t(IndCannPropODB)  
#view(transposed\_IndCannPropLemonM)  
  
transposed\_IndCannPropODB\_melt <- melt(IndCannPropODB, id.vars = "Description")  
  
#Change the column Variable to Cannabinoids.   
colnames(transposed\_IndCannPropODB\_melt)[colnames(transposed\_IndCannPropODB\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropODB\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 8 rows containing missing values (`position\_stack()`).

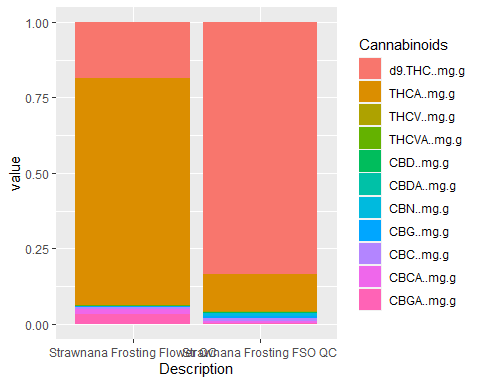


SNN <-   
 simpleChem %>%  
 filter(grepl("Strawnana Frosting", Description)) %>%  
 group\_by(Description)  
  
SNN <- SNN %>%  
 filter(Description != "Strawnana Frosting Live FSO")  
View(SNN)  
  
SNNIndCannProp <- SNN%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
#view(SNNIndCannProp)  
  
IndCannPropSNN <- SNNIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

#view(IndCannPropSNN)  
  
colnames(IndCannPropSNN) <- gsub(".\_IndCann", "", colnames(IndCannPropSNN))  
  
  
transposed\_IndCannPropSNN <- t(IndCannPropSNN)  
#view(transposed\_IndCannPropSNN)  
  
transposed\_IndCannPropSNN\_melt <- melt(IndCannPropSNN, id.vars = "Description")  
  
#Change the column Variable to Cannabinoids.   
colnames(transposed\_IndCannPropSNN\_melt)[colnames(transposed\_IndCannPropSNN\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropSNN\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 8 rows containing missing values (`position\_stack()`).

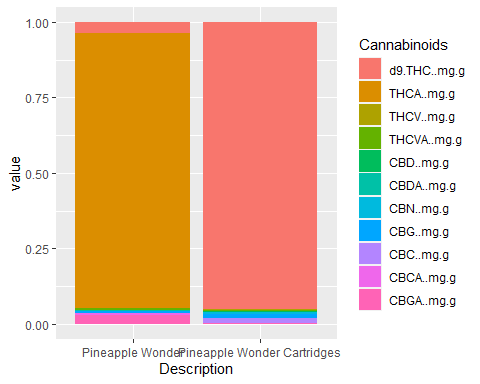


PW <-   
 simpleChem %>%  
 filter(grepl("Pineapple Wonder", Description)) %>%  
 group\_by(Description)  
  
  
PW <- PW %>%  
 filter(Description != "Pineapple Wonder THC Tincture 30 mL",  
 Description != "Pineapple Wonder FSO",  
 Description != "Pineapple Wonder Cartridge"  
 )  
  
PWIndCannProp <- PW%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
#view(PWIndCannProp)  
  
IndCannPropPW <- PWIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

#view(IndCannPropPW)  
  
colnames(IndCannPropPW) <- gsub(".\_IndCann", "", colnames(IndCannPropPW))  
  
  
transposed\_IndCannPropPW <- t(IndCannPropPW)  
#view(transposed\_IndCannPropSNN)  
  
transposed\_IndCannPropPW\_melt <- melt(IndCannPropPW, id.vars = "Description")  
  
#Change the column Variable to Cannabinoids.   
colnames(transposed\_IndCannPropPW\_melt)[colnames(transposed\_IndCannPropPW\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropPW\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 10 rows containing missing values (`position\_stack()`).

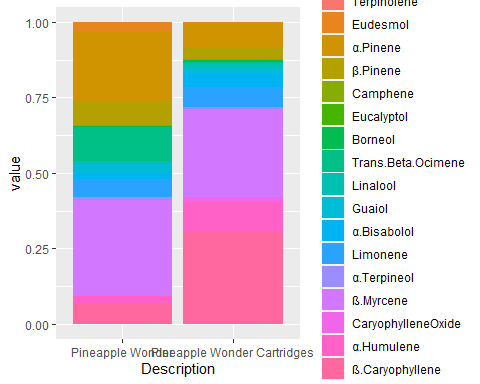


PW <-   
 simpleChem %>%  
 filter(grepl("Pineapple Wonder", Description)) %>%  
 group\_by(Description)  
  
  
PW <- PW %>%  
 filter(Description != "Pineapple Wonder THC Tincture 30 mL",  
 Description != "Pineapple Wonder FSO",  
 Description != "Pineapple Wonder Cartridge"  
 )  
  
PWIndTerpProp <- PW%>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropPW <- PWIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

#Code to remove ...\_IndTerp from the individual terpenes in the transposed dataframe.   
  
colnames(IndTerpPropPW) <- gsub("....\_IndTerp", "", colnames(IndTerpPropPW))  
  
transposed\_IndTerpPropPW <- t(IndTerpPropPW)  
  
transposed\_IndTerpPropPW\_melt <- melt(IndTerpPropPW, id.vars = "Description")  
  
#Change the column Variable to Terpenes.   
colnames(transposed\_IndTerpPropPW\_melt)[colnames(transposed\_IndTerpPropPW\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropPW\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 25 rows containing missing values (`position\_stack()`).

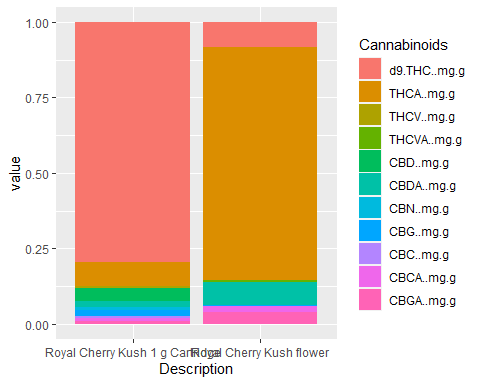


#Royal Cherry Kush Cannabinoid analysis  
RCK <-   
 simpleChem %>%  
 filter(grepl("Royal Cherry Kush", Description)) %>%  
 group\_by(Description)  
  
RCK <- RCK %>%  
 filter(Description != "Royal Cherry Kush FSO QC",  
 Lot.Batch.. != "MGRCK210615",  
 )  
  
view(RCK)  
RCKIndCannProp <- RCK%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
IndCannPropRCK <- RCKIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropRCK) <- gsub(".\_IndCann", "", colnames(IndCannPropRCK))  
  
  
transposed\_IndCannPropRCK <- t(IndCannPropRCK)  
#view(transposed\_IndCannPropRCK)  
  
transposed\_IndCannPropRCK\_melt <- melt(IndCannPropRCK, id.vars = "Description")  
  
#Change the column Variable to Cannabinoids.   
colnames(transposed\_IndCannPropRCK\_melt)[colnames(transposed\_IndCannPropRCK\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropRCK\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 9 rows containing missing values (`position\_stack()`).

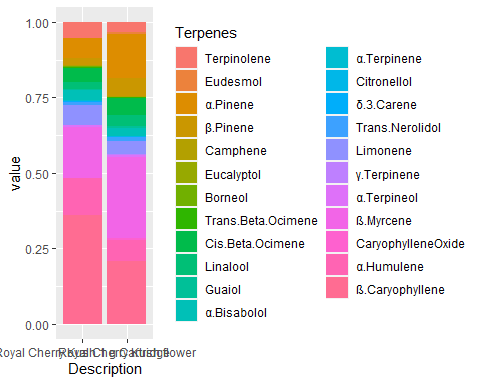


#Royal Cherry Kush Terpene proportions  
  
RCK <-   
 simpleChem %>%  
 filter(grepl("Royal Cherry Kush", Description)) %>%  
 group\_by(Description)  
  
RCK <- RCK %>%  
 filter(Description != "Royal Cherry Kush FSO QC",  
 Lot.Batch.. != "MGRCK210615",  
 )  
RCKIndTerpProp <- RCK%>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropRCK <- RCKIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

#Code to remove ...\_IndTerp from the individual terpenes in the transposed dataframe.   
  
colnames(IndTerpPropRCK) <- gsub("....\_IndTerp", "", colnames(IndTerpPropRCK))  
  
transposed\_IndTerpPropRCK <- t(IndTerpPropRCK)  
  
transposed\_IndTerpPropRCK\_melt <- melt(IndTerpPropRCK, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropRCK\_melt)[colnames(transposed\_IndTerpPropRCK\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropRCK\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 12 rows containing missing values (`position\_stack()`).

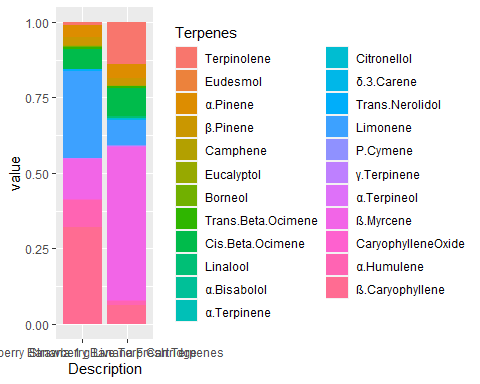


#Only Terpene analysis Being done on Strawberry banana.   
 SBN <-   
 simpleChem %>%  
 filter(grepl("Strawberry Banana", Description)) %>%  
 group\_by(Description)  
#view(SBN)  
  
SBNIndTerpProp <- SBN%>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropSBN <- SBNIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

#Code to remove ...\_IndTerp from the individual terpenes in the transposed dataframe.   
  
colnames(IndTerpPropSBN) <- gsub("....\_IndTerp", "", colnames(IndTerpPropSBN))  
  
transposed\_IndTerpPropSBN <- t(IndTerpPropSBN)  
  
transposed\_IndTerpPropSBN\_melt <- melt(IndTerpPropSBN, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropSBN\_melt)[colnames(transposed\_IndTerpPropSBN\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropSBN\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 15 rows containing missing values (`position\_stack()`).

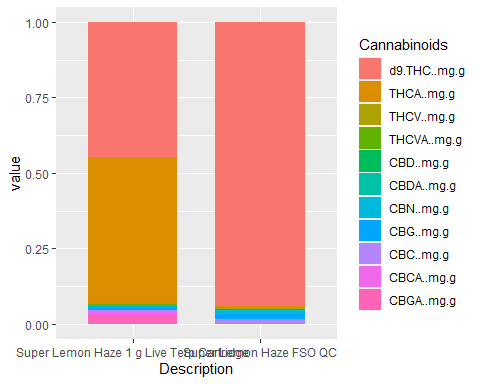


#Super Lemon Haze Cannabinoid Proportions   
 SLH <-   
 simpleChem %>%  
 filter(grepl("Super Lemon Haze", Description)) %>%  
 group\_by(Description)  
  
  
SLH <- SLH %>%  
 filter(Description != "Super Lemon Haze Fresh Terpenes",  
 Description != "Super Lemon Haze FSO QC 01/13/22"  
 )  
view(SLH)  
  
#Orange Durban Cannabinoid Proportion formulas for a comparison stacked bar chart.   
SLHIndCannProp <- SLH%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
IndCannPropSLH <- SLHIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropSLH) <- gsub(".\_IndCann", "", colnames(IndCannPropSLH))  
  
  
transposed\_IndCannPropSLH <- t(IndCannPropSLH)  
  
transposed\_IndCannPropSLH\_melt <- melt(IndCannPropSLH, id.vars = "Description")  
  
colnames(transposed\_IndCannPropSLH\_melt)[colnames(transposed\_IndCannPropSLH\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropSLH\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack", width = 0.7)

## Warning: Removed 5 rows containing missing values (`position\_stack()`).

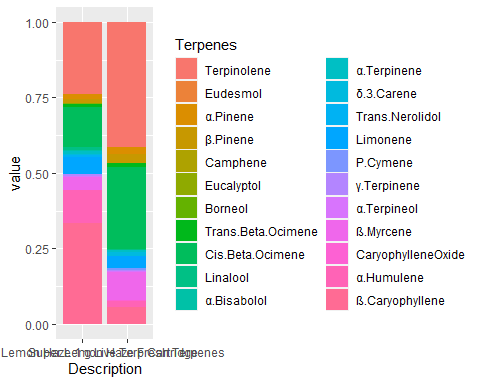


#Super Lemon Haze Live Terpenes compared to SLH Live terpene cartridge.   
 SLHT <-   
 simpleChem %>%  
 filter(grepl("Super Lemon Haze", Description)) %>%  
 group\_by(Description)  
  
SLHT <- SLHT %>%  
 filter(Description != "Super Lemon Haze FSO QC",  
 Description != "Super Lemon Haze FSO QC 01/13/22"  
 )  
view(SLHT)  
  
  
SLHTIndTerpProp <- SLHT %>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropSLHT <- SLHTIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

#Code to remove ...\_IndTerp from the individual terpenes in the transposed dataframe.   
  
colnames(IndTerpPropSLHT) <- gsub("....\_IndTerp", "", colnames(IndTerpPropSLHT))  
  
transposed\_IndTerpPropSLHT <- t(IndTerpPropSLHT)  
  
transposed\_IndTerpPropSLHT\_melt <- melt(IndTerpPropSLHT, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropSLHT\_melt)[colnames(transposed\_IndTerpPropSLHT\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropSLHT\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 15 rows containing missing values (`position\_stack()`).

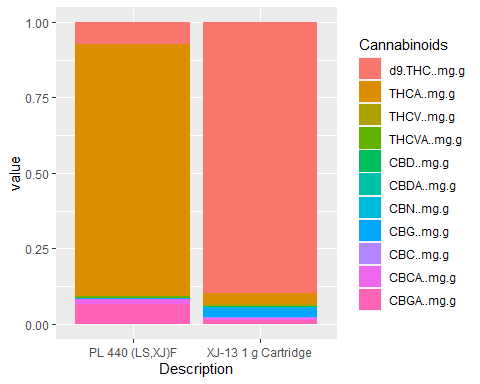


#XJ13 Cannabinoid Analysis   
 XJ <-   
 simpleChem %>%  
 filter(grepl("XJ", Description)) %>%  
 group\_by(Description)  
  
view(XJ)  
  
XJ <- XJ %>%  
 filter(Description != "PL 440 (LS, XJ)CRUDE",  
 Description != "INFUSED XJ-13 FSO",  
 Description != "CHM-LSXJ190826-FSO-QC-02",  
 Description != "CHM-LSXJ190826-FSO-QC",  
 Description != "XJ-13 Uno Pod Vape Cartridge",  
 Description != "XJ-13 Terpenes",  
 Description != "XJ-13 Pod"  
 )  
  
#view(XJ)  
  
XJIndCannProp <- XJ%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
IndCannPropXJ <- XJIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropXJ) <- gsub(".\_IndCann", "", colnames(IndCannPropXJ))  
  
  
transposed\_IndCannPropXJ <- t(IndCannPropXJ)  
  
transposed\_IndCannPropXJ\_melt <- melt(IndCannPropXJ, id.vars = "Description")  
  
colnames(transposed\_IndCannPropXJ\_melt)[colnames(transposed\_IndCannPropXJ\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropXJ\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 10 rows containing missing values (`position\_stack()`).

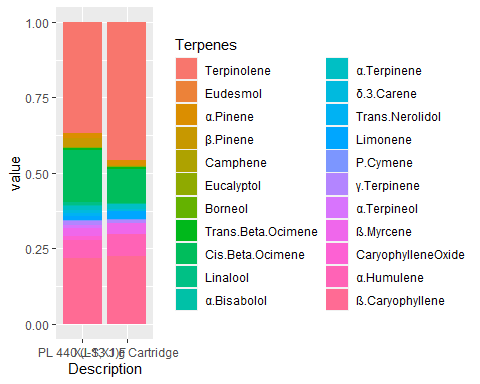


#XJ13 Terpene Analysis   
 XJT <-   
 simpleChem %>%  
 filter(grepl("XJ", Description)) %>%  
 group\_by(Description)  
  
view(XJT)  
  
XJT <- XJT %>%  
 filter(Description != "PL 440 (LS, XJ)CRUDE",  
 Description != "INFUSED XJ-13 FSO",  
 Description != "CHM-LSXJ190826-FSO-QC-02",  
 Description != "CHM-LSXJ190826-FSO-QC",  
 Description != "XJ-13 Uno Pod Vape Cartridge",  
 Description != "XJ-13 Terpenes",  
 Description != "XJ-13 Pod"  
 )  
  
XJTIndTerpProp <- XJT %>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropXJT <- XJTIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

colnames(IndTerpPropXJT) <- gsub("....\_IndTerp", "", colnames(IndTerpPropXJT))  
  
transposed\_IndTerpPropXJT <- t(IndTerpPropXJT)  
  
transposed\_IndTerpPropXJT\_melt <- melt(IndTerpPropXJT, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropXJT\_melt)[colnames(transposed\_IndTerpPropXJT\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropXJT\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 17 rows containing missing values (`position\_stack()`).

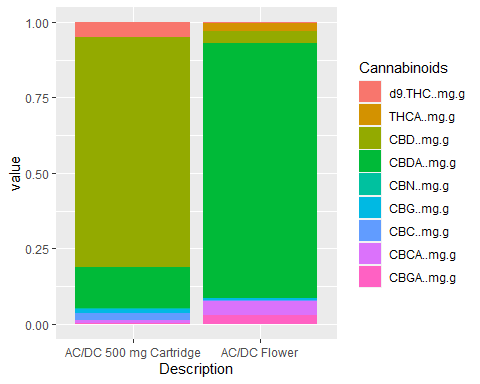


#AC/DC Cannabinoid Analysis  
 ACD <-   
 simpleChem %>%  
 filter(grepl("AC/DC", Description)) %>%  
 group\_by(Description)  
  
ACD <- ACD %>%  
 filter(Description != "AC/DC Tincture QC",  
 Description != "AC/DC Tincture",  
 Description != "AC/DC 30 mL CBD Tincture",  
 Description != "AC/DC CBD 30 mL Tincture",  
 Description != "AC/DC Tincture (30ml)",  
 Description != "Uninfused AC/DC FSO",  
 Lot.Batch.. != "FRFACD210910"  
 )  
ACDIndCannProp <- ACD%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
  
IndCannPropACD <- ACDIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropACD) <- gsub(".\_IndCann", "", colnames(IndCannPropACD))  
  
  
transposed\_IndCannPropACD <- t(IndCannPropACD)  
  
transposed\_IndCannPropACD\_melt <- melt(IndCannPropACD, id.vars = "Description")  
  
colnames(transposed\_IndCannPropACD\_melt)[colnames(transposed\_IndCannPropACD\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropACD\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 9 rows containing missing values (`position\_stack()`).

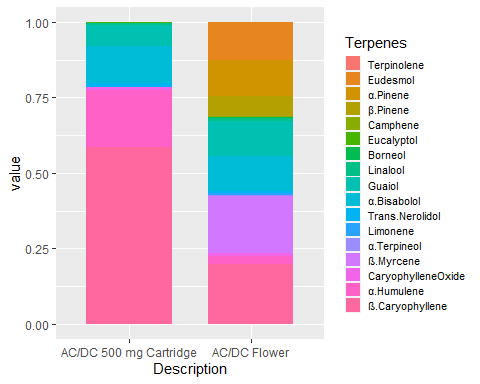


ACDT <-   
 simpleChem %>%  
 filter(grepl("AC/DC", Description)) %>%  
 group\_by(Description)  
  
ACDT <- ACDT %>%  
 filter(Description != "AC/DC Tincture QC",  
 Description != "AC/DC Tincture",  
 Description != "AC/DC 30 mL CBD Tincture",  
 Description != "AC/DC CBD 30 mL Tincture",  
 Description != "AC/DC Tincture (30ml)",  
 Description != "Uninfused AC/DC FSO",  
 Lot.Batch.. != "FRFACD210910"  
 )  
  
  
ACDTIndTerpProp <- ACDT %>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropACDT <- ACDTIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

colnames(IndTerpPropACDT) <- gsub("....\_IndTerp", "", colnames(IndTerpPropACDT))  
  
transposed\_IndTerpPropACDT <- t(IndTerpPropACDT)  
  
transposed\_IndTerpPropACDT\_melt <- melt(IndTerpPropACDT, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropACDT\_melt)[colnames(transposed\_IndTerpPropACDT\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropACDT\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack", width = 0.7)+  
 theme(legend.text = element\_text(size = 8))+  
 theme(legend.key.size = unit(4, "mm"))

## Warning: Removed 29 rows containing missing values (`position\_stack()`).

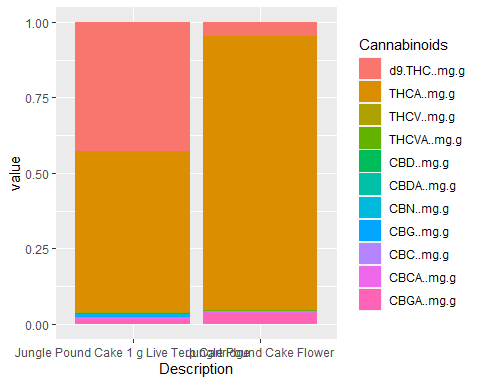


#Jungle Pound Cake Cannabinoid Only Analysis  
 JPC <-   
 simpleChem %>%  
 filter(grepl("Jungle Pound Cake", Description)) %>%  
 group\_by(Description)  
  
JPC <- JPC %>%  
 filter(Description != "Jungle Pound Cake FSO")  
  
  
JPCIndCannProp <- JPC%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
  
IndCannPropJPC <- JPCIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropJPC) <- gsub(".\_IndCann", "", colnames(IndCannPropJPC))  
  
  
transposed\_IndCannPropJPC <- t(IndCannPropJPC)  
  
transposed\_IndCannPropJPC\_melt <- melt(IndCannPropJPC, id.vars = "Description")  
  
colnames(transposed\_IndCannPropJPC\_melt)[colnames(transposed\_IndCannPropJPC\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropJPC\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 10 rows containing missing values (`position\_stack()`).

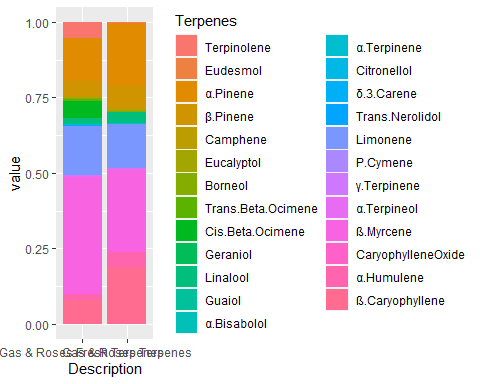


#Gas and Roses Fresh Vs. Cured Terpene analysis  
 GNR <-   
 simpleChem %>%  
 filter(grepl("Gas & Roses", Description)) %>%  
 group\_by(Description)  
  
GNR <- GNR %>%  
 filter(Description != "Gas & Roses Trim QC",  
 Description != "Gas & Roses FSO QC",  
 Description != "Gas & Roses Live Drip Diamonds 1 g"  
 )  
view(GNR)  
  
GNRIndTerpProp <- GNR %>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropGNR <- GNRIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

colnames(IndTerpPropGNR) <- gsub("....\_IndTerp", "", colnames(IndTerpPropGNR))  
  
transposed\_IndTerpPropGNR <- t(IndTerpPropGNR)  
  
transposed\_IndTerpPropGNR\_melt <- melt(IndTerpPropGNR, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropGNR\_melt)[colnames(transposed\_IndTerpPropGNR\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropGNR\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack")

## Warning: Removed 10 rows containing missing values (`position\_stack()`).

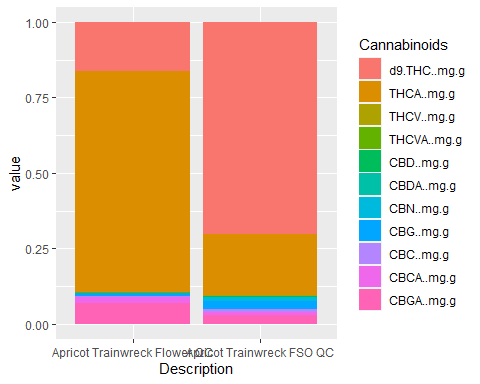


#Apricot Trainwreck Cannabinoid Analysis  
 APW <-   
 simpleChem %>%  
 filter(grepl("Apricot Trainwreck", Description)) %>%  
 group\_by(Description)  
view(APW)  
  
APWIndCannProp <- APW%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
  
IndCannPropAPW <- APWIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropAPW) <- gsub(".\_IndCann", "", colnames(IndCannPropAPW))  
  
  
transposed\_IndCannPropAPW <- t(IndCannPropAPW)  
  
transposed\_IndCannPropAPW\_melt <- melt(IndCannPropAPW, id.vars = "Description")  
  
colnames(transposed\_IndCannPropAPW\_melt)[colnames(transposed\_IndCannPropAPW\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropAPW\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 9 rows containing missing values (`position\_stack()`).

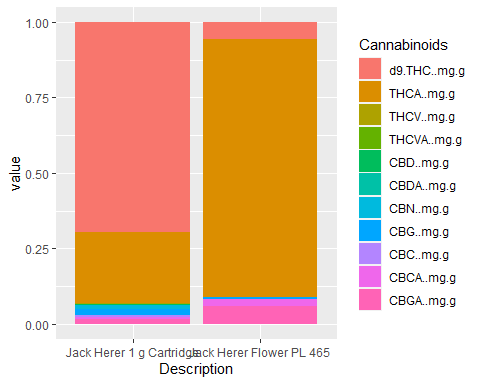


#Apricot Trainwreck Cannabinoid Analysis  
 Jack <-   
 simpleChem %>%  
 filter(grepl("Jack", Description)) %>%  
 group\_by(Description)  
  
Jack <- Jack %>%  
 filter(Lot.Batch.. != "MMJCT210811",  
 Lot.Batch.. != "MMJHR210614",  
 Lot.Batch.. != "FRFJHR220915-1G",  
 Lot.Batch.. != "CHM-MMJHR200824-FSO",  
 Lot.Batch.. != "CHM-FRFJHR220329-FSO",  
 Lot.Batch.. != "CHM-MMJCT210712-FSO",  
 Lot.Batch.. != "CHM-FRFJHR220329-FSO",  
 Lot.Batch.. != "CHM-MMJHR200824-FSO",  
 )  
  
#view(Jack)  
  
JackIndCannProp <- Jack%>%  
 mutate\_at(vars(d9.THC..mg.g.:CBGA..mg.g.),list(IndCann=~./total\_cannabinoidsMg.g))  
  
  
IndCannPropJack <- JackIndCannProp %>%   
 select(52:64)

## Adding missing grouping variables: `Description`

colnames(IndCannPropJack) <- gsub(".\_IndCann", "", colnames(IndCannPropJack))  
  
  
transposed\_IndCannPropJack <- t(IndCannPropJack)  
  
transposed\_IndCannPropJack\_melt <- melt(IndCannPropJack, id.vars = "Description")  
  
colnames(transposed\_IndCannPropJack\_melt)[colnames(transposed\_IndCannPropJack\_melt) == "variable"] <- "Cannabinoids"  
  
ggplot(data = transposed\_IndCannPropJack\_melt, aes(x = Description, y = value, fill = Cannabinoids)) +  
 geom\_col(position = "stack")

## Warning: Removed 10 rows containing missing values (`position\_stack()`).

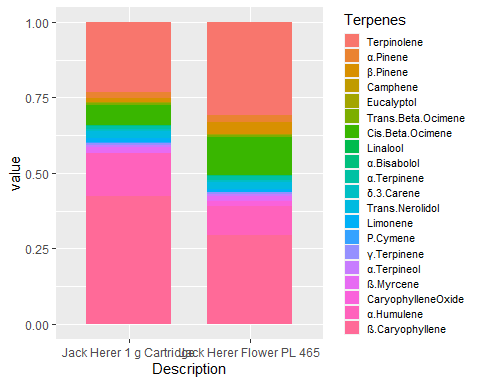


#Jack Herer Terpene Analysis  
 JT <-   
 simpleChem %>%  
 filter(grepl("Jack", Description)) %>%  
 group\_by(Description)  
  
JT <- JT %>%  
 filter(Lot.Batch.. != "MMJCT210811",  
 Lot.Batch.. != "MMJHR210614",  
 Lot.Batch.. != "FRFJHR220915-1G",  
 Lot.Batch.. != "CHM-MMJHR200824-FSO",  
 Lot.Batch.. != "CHM-FRFJHR220329-FSO",  
 Lot.Batch.. != "CHM-MMJCT210712-FSO",  
 Lot.Batch.. != "CHM-FRFJHR220329-FSO",  
 Lot.Batch.. != "CHM-MMJHR200824-FSO",  
 )  
  
JTIndTerpProp <- JT %>%  
 mutate\_at(vars(Terpinolene....:ß.Caryophyllene....),list(IndTerp=~./total\_terpsMg.g))  
  
IndTerpPropJT <- JTIndTerpProp %>%   
 select(52:79)

## Adding missing grouping variables: `Description`

colnames(IndTerpPropJT) <- gsub("....\_IndTerp", "", colnames(IndTerpPropJT))  
  
transposed\_IndTerpPropJT <- t(IndTerpPropJT)  
  
transposed\_IndTerpPropJT\_melt <- melt(IndTerpPropJT, id.vars = "Description")  
  
colnames(transposed\_IndTerpPropJT\_melt)[colnames(transposed\_IndTerpPropJT\_melt) == "variable"] <- "Terpenes"  
  
ggplot(data = transposed\_IndTerpPropJT\_melt, aes(x = Description, y = value, fill = Terpenes)) +  
 geom\_col(position = "stack", width = 0.7)+  
 theme(legend.text = element\_text(size = 8))+  
 theme(legend.key.size = unit(4, "mm"))

## Warning: Removed 19 rows containing missing values (`position\_stack()`).



```