H7F5633

Anton Hellberg

2025-03-14

***Task 4 - R basic operations***

**1. What is the square root of 10?**

sqrt(10)

## [1] 3.162278

**2. What is the logarithm of 32 to the base 2?**

log2(32)

## [1] 5

**3. What is the sum of the numbers from 1 to 1000?**

sum(1:1000)

## [1] 500500

**4. What is the sum of all even numbers from 2 to 1000?**

sum(seq(2, 1000, by=2))

## [1] 250500

**5. How many pairwise comparisons are there for 100 genes?**

100\*99/2

## [1] 4950

**6. And how many ways to arrange 100 genes in triples?**

100\*99\*98/6

## [1] 161700

***Task 5 - Using R example datasets***

**1. Use the R internal CO2 dataset (“data(CO2)”).**

**2. Describe briefly the content of the CO2 dataset using the help function.**

*Answer: The CO2 data frame has 84 rows and 5 columns of data from an experiment on the cold tolerance of the grass species Echinochloa crus-galli. The columns are: Plant, Type, Treatment, conc, uptake.*

**3. What is the average and median CO2 uptake of the plants from Quebec and Mississippi?**

mean(CO2$uptake[CO2$Type == "Mississippi"])

## [1] 20.88333

median(CO2$uptake[CO2$Type == "Mississippi"])

## [1] 19.3

mean(CO2$uptake[CO2$Type == "Quebec"])

## [1] 33.54286

median(CO2$uptake[CO2$Type == "Quebec"])

## [1] 37.15

**4. [Optional] In the “airway” example data from Bioconductor, how many genes are expressed in each sample? How many genes are not expressed in any sample?**

*I could not complete this as Bioconductor was unable to be installed, I will try to fix this the coming days.*

***Task 6 - R Functions***

**1. Write a function that calculates the ratio of the mean and the median of a given vector. This is a helpful measure to detect data with outlying values. Note: See Reference for R language**

*answer: mean(x)/median(x)*

**2. Write a function that ignores the lowest and the highest value from a given vector and calculate the mean.** *answer: mean(x[-c(min(x), max(x))])*

**3. Read about piping from here:**[**https://r4ds.had.co.nz/pipes.html#pipes**](https://r4ds.had.co.nz/pipes.html#pipes) **(you don’t have to learn everything, a basic understanding of the usage is enough). Write a short (max. 300 characters, no spaces) explanation of why, how, and when not to use pipes.**

*Answer: For example if I were to create a plot using ggplot. Creating one long pipe could make debugging the code a challenge. Therefore it would be better to subdivide the ggplot into functioning parts and applying layers to add more features. So in essence, if you will have a long pipe, it is better to choose an alterantive.*

**4. Familiarize yourself with the apply-family of functions (apply, lapply, sapply etc.)** [**http://uc-r.github.io/apply\_family**](http://uc-r.github.io/apply_family) **Write a short explanation (max. 300 characters, no spaces) of why they could be useful in your work.**

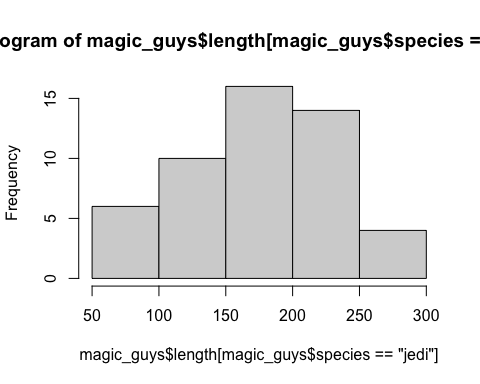
*Answer: Apply can be used as a fast means to understand the data, such as quickly examining means or medians. And as I used lapply later in this assigment, it is good for repetetive tasks such as transforming data or applying models. In summary, apply is suitable for fast actions and also for actions that are repetetive.*

***Task 7 - Basic visualization with R***

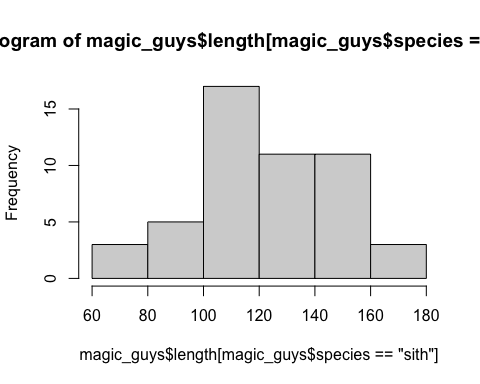
magic\_guys <- read.csv("~/Library/CloudStorage/OneDrive-KarolinskaInstitutet/Doktoreringen/22. R/Projekt/Bioinformatics Analysis and Visualisation of Medical Genomics Data - VT2025/magic\_guys.csv")

**1. Compare the distributions of the body heights of the two species from the ‘magic\_guys.csv’ dataset graphically**

hist(magic\_guys$length[magic\_guys$species == "jedi"])



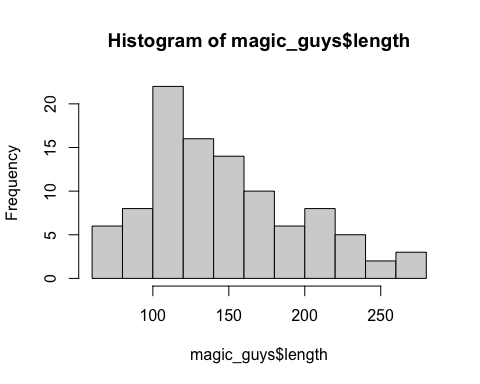
hist(magic\_guys$length[magic\_guys$species == "sith"])



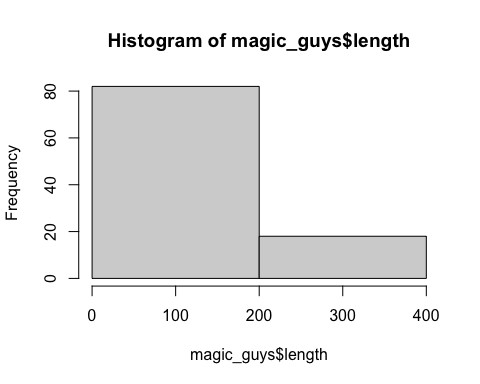
## Histograms are primarily used for distribution, this command lets you choose to see the distribution of heights and also specifically the two separate [magic\_guys$species == "sith"] species. ##

**a. using the basic ‘hist’ function as well as ‘ggplot’ and ‘geom\_histogram’ functions from the ggplot2 package. Optimize the plots for example by trying several different ‘breaks’. Note that ggplot2-based functions give you many more options for changing the visualization parameters, try some of them.**

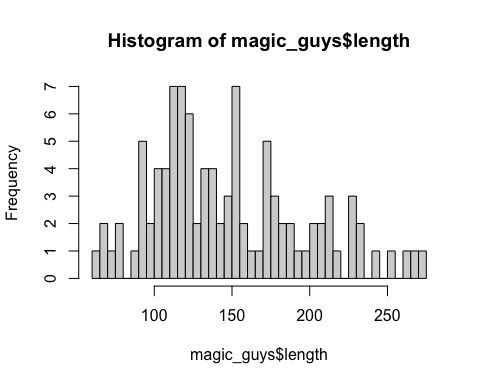
hist(magic\_guys$length)



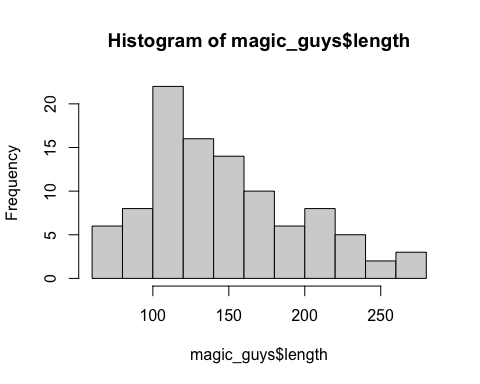
hist(magic\_guys$length, breaks = 1) ## only specifies approximately how many bins I want



hist(magic\_guys$length, breaks = 50)## only specifies approximately how many bins I want



hist(magic\_guys$length, breaks = 10)## only specifies approximately how many bins I want



#I deem 10 breaks to be the most optimal to correctly visualize the distribution of lengths  
  
library(ggplot2)  
library(tidyr)  
library(dplyr)

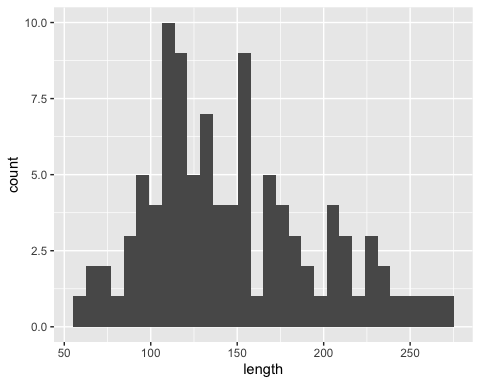
##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

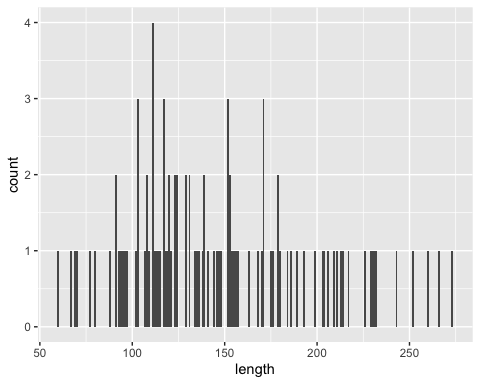
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

magic\_guys %>%  
ggplot(aes(x=length)) +  
geom\_histogram()

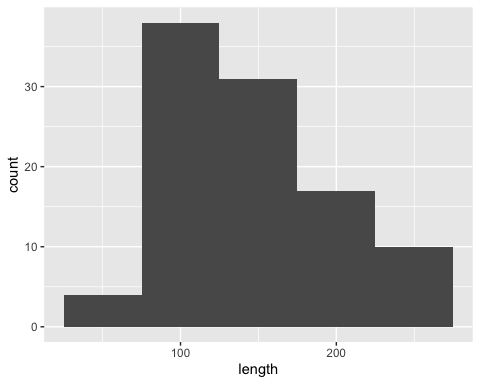
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



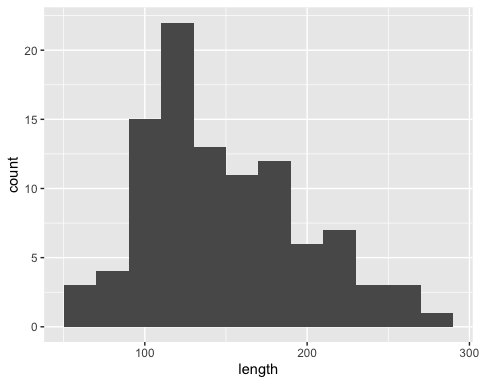
magic\_guys %>%  
ggplot(aes(x=length)) +  
geom\_histogram(binwidth=1) #Tells the width of each bin, so reverse to the hist(x, breaks = y) function



magic\_guys %>%  
ggplot(aes(x=length)) +  
geom\_histogram(binwidth=50) #Tells the width of each bin, so reverse to the hist(x, breaks = y) function



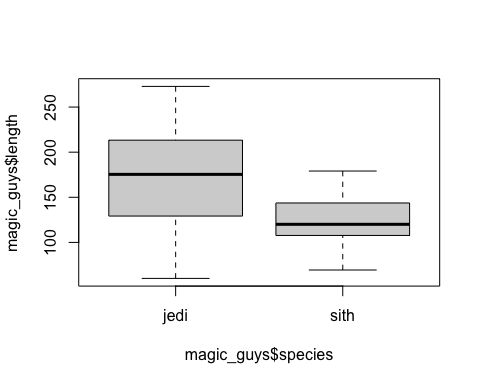
magic\_guys %>%  
ggplot(aes(x=length)) +  
geom\_histogram(binwidth=20)



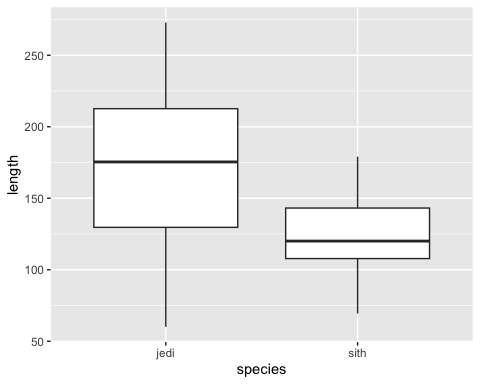
#I deem a binwidth of 20 is best for vizualizing the distribution of the data with ggplot

**b. Do the same comparison as in a. but with boxplots. If you want to use the ggplot2-package, use the functions ‘ggplot’ and ‘geom\_boxplot’.**

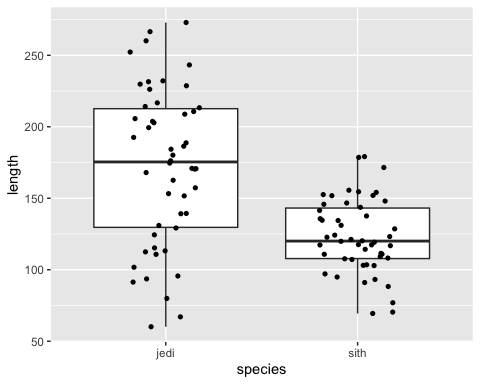
boxplot(magic\_guys$length ~ magic\_guys$species)



magic\_guys %>%  
ggplot(aes(x=species, y=length)) +  
geom\_boxplot()



magic\_guys %>%  
ggplot(aes(x=species, y=length)) +  
 geom\_boxplot() +  
 geom\_jitter(shape=16, position=position\_jitter(0.2))



#This gives the boxplot a good visual overview, both showing the median, IQR and extremes and also the various data points to see how the data is distributed, although the aestethics could be improved.

**c. Save the plots with the ‘png’,**

#example  
png(file="/Users/antonh/desktop/example.png",  
width=600, height=350)  
hist(magic\_guys$length)  
dev.off()

## quartz\_off\_screen   
## 2

**2. Load the gene expression data matrix from the ‘microarray\_data.tab’ dataset provided in the shared folder, it is a big tabular separated matrix.**

microarray\_data <- read.delim("~/Library/CloudStorage/OneDrive-KarolinskaInstitutet/Doktoreringen/22. R/Projekt/Bioinformatics Analysis and Visualisation of Medical Genomics Data - VT2025/Bioinformatics H7F5633 VT 2025/microarray\_data.tab")

**a. How big is the matrix in terms of rows and columns?**

nrow(microarray\_data) #gives the row numbers

## [1] 553

ncol(microarray\_data) #gives column numbers

## [1] 1000

#or  
dim(microarray\_data) #gives the dimension of the data frame; rows and columns.

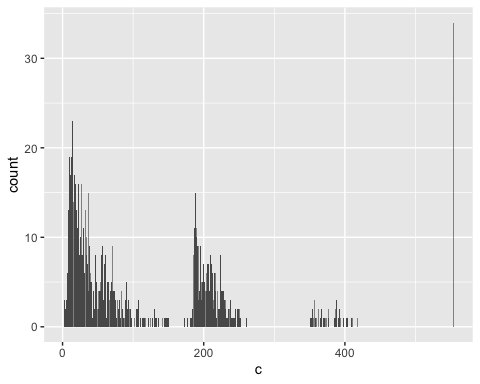
## [1] 553 1000

**b. Count the missing values per gene and visualize this result.**

colSums(is.na(microarray\_data))

## g1 g2 g3 g4 g5 g6 g7 g8 g9 g10 g11 g12 g13   
## 130 104 74 93 81 30 31 26 12 56 72 105 55   
## g14 g15 g16 g17 g18 g19 g20 g21 g22 g23 g24 g25 g26   
## 133 75 67 21 553 13 9 207 190 96 141 122 149   
## g27 g28 g29 g30 g31 g32 g33 g34 g35 g36 g37 g38 g39   
## 47 59 210 22 14 52 14 24 70 83 64 91 144   
## g40 g41 g42 g43 g44 g45 g46 g47 g48 g49 g50 g51 g52   
## 124 216 64 13 70 61 108 68 404 106 91 230 231   
## g53 g54 g55 g56 g57 g58 g59 g60 g61 g62 g63 g64 g65   
## 207 195 553 196 186 553 224 362 198 217 208 210 188   
## g66 g67 g68 g69 g70 g71 g72 g73 g74 g75 g76 g77 g78   
## 351 187 189 187 194 185 185 354 203 216 216 245 201   
## g79 g80 g81 g82 g83 g84 g85 g86 g87 g88 g89 g90 g91   
## 553 189 188 224 553 213 188 194 192 213 219 229 365   
## g92 g93 g94 g95 g96 g97 g98 g99 g100 g101 g102 g103 g104   
## 223 369 353 195 230 250 238 553 204 228 199 192 211   
## g105 g106 g107 g108 g109 g110 g111 g112 g113 g114 g115 g116 g117   
## 373 251 261 207 362 187 200 209 210 224 203 205 189   
## g118 g119 g120 g121 g122 g123 g124 g125 g126 g127 g128 g129 g130   
## 212 236 260 33 26 25 22 14 48 36 42 18 194   
## g131 g132 g133 g134 g135 g136 g137 g138 g139 g140 g141 g142 g143   
## 213 367 198 199 553 185 553 553 191 186 14 177 8   
## g144 g145 g146 g147 g148 g149 g150 g151 g152 g153 g154 g155 g156   
## 13 17 8 186 202 21 30 198 186 207 185 188 196   
## g157 g158 g159 g160 g161 g162 g163 g164 g165 g166 g167 g168 g169   
## 215 186 188 187 36 27 28 18 212 36 8 23 33   
## g170 g171 g172 g173 g174 g175 g176 g177 g178 g179 g180 g181 g182   
## 23 226 376 200 215 193 214 185 224 203 193 10 9   
## g183 g184 g185 g186 g187 g188 g189 g190 g191 g192 g193 g194 g195   
## 10 18 32 47 31 44 15 20 36 35 36 61 20   
## g196 g197 g198 g199 g200 g201 g202 g203 g204 g205 g206 g207 g208   
## 188 12 8 12 187 12 7 19 215 15 12 8 11   
## g209 g210 g211 g212 g213 g214 g215 g216 g217 g218 g219 g220 g221   
## 23 173 9 14 7 29 15 36 15 10 16 10 56   
## g222 g223 g224 g225 g226 g227 g228 g229 g230 g231 g232 g233 g234   
## 31 83 52 27 27 12 13 38 44 17 24 188 11   
## g235 g236 g237 g238 g239 g240 g241 g242 g243 g244 g245 g246 g247   
## 36 30 39 55 64 38 81 66 65 61 24 26 47   
## g248 g249 g250 g251 g252 g253 g254 g255 g256 g257 g258 g259 g260   
## 26 36 23 47 203 2 20 17 8 7 70 34 392   
## g261 g262 g263 g264 g265 g266 g267 g268 g269 g270 g271 g272 g273   
## 31 49 67 60 38 36 18 61 38 32 12 14 15   
## g274 g275 g276 g277 g278 g279 g280 g281 g282 g283 g284 g285 g286   
## 61 24 22 8 3 26 89 67 18 27 92 59 65   
## g287 g288 g289 g290 g291 g292 g293 g294 g295 g296 g297 g298 g299   
## 65 57 53 410 17 97 15 60 68 106 127 64 81   
## g300 g301 g302 g303 g304 g305 g306 g307 g308 g309 g310 g311 g312   
## 39 388 34 10 25 14 18 40 55 57 47 92 61   
## g313 g314 g315 g316 g317 g318 g319 g320 g321 g322 g323 g324 g325   
## 82 89 70 83 11 15 30 98 115 74 57 90 54   
## g326 g327 g328 g329 g330 g331 g332 g333 g334 g335 g336 g337 g338   
## 53 72 51 553 19 91 147 113 116 130 60 84 47   
## g339 g340 g341 g342 g343 g344 g345 g346 g347 g348 g349 g350 g351   
## 70 34 18 23 25 150 48 54 56 58 46 33 238   
## g352 g353 g354 g355 g356 g357 g358 g359 g360 g361 g362 g363 g364   
## 553 220 200 388 216 218 211 212 205 192 370 371 208   
## g365 g366 g367 g368 g369 g370 g371 g372 g373 g374 g375 g376 g377   
## 200 191 240 553 186 248 50 65 45 71 20 29 74   
## g378 g379 g380 g381 g382 g383 g384 g385 g386 g387 g388 g389 g390   
## 93 239 59 212 205 409 204 207 225 241 358 553 553   
## g391 g392 g393 g394 g395 g396 g397 g398 g399 g400 g401 g402 g403   
## 402 57 53 55 45 211 22 47 39 136 208 200 191   
## g404 g405 g406 g407 g408 g409 g410 g411 g412 g413 g414 g415 g416   
## 210 248 367 195 196 196 210 57 29 84 40 117 62   
## g417 g418 g419 g420 g421 g422 g423 g424 g425 g426 g427 g428 g429   
## 553 210 91 41 75 13 69 33 59 55 24 32 94   
## g430 g431 g432 g433 g434 g435 g436 g437 g438 g439 g440 g441 g442   
## 79 553 359 243 185 190 215 206 230 205 391 6 8   
## g443 g444 g445 g446 g447 g448 g449 g450 g451 g452 g453 g454 g455   
## 47 17 132 12 30 17 37 195 17 9 86 30 202   
## g456 g457 g458 g459 g460 g461 g462 g463 g464 g465 g466 g467 g468   
## 10 20 33 63 61 392 356 195 190 220 229 188 189   
## g469 g470 g471 g472 g473 g474 g475 g476 g477 g478 g479 g480 g481   
## 205 245 10 28 37 6 21 190 70 9 110 23 87   
## g482 g483 g484 g485 g486 g487 g488 g489 g490 g491 g492 g493 g494   
## 29 80 30 14 23 14 11 23 33 215 187 201 190   
## g495 g496 g497 g498 g499 g500 g501 g502 g503 g504 g505 g506 g507   
## 188 221 224 553 190 192 39 70 10 9 36 29 64   
## g508 g509 g510 g511 g512 g513 g514 g515 g516 g517 g518 g519 g520   
## 9 13 189 17 16 184 72 214 43 39 217 553 190   
## g521 g522 g523 g524 g525 g526 g527 g528 g529 g530 g531 g532 g533   
## 48 200 4 63 61 27 553 36 22 73 553 387 197   
## g534 g535 g536 g537 g538 g539 g540 g541 g542 g543 g544 g545 g546   
## 192 209 207 238 553 185 201 56 41 14 189 8 12   
## g547 g548 g549 g550 g551 g552 g553 g554 g555 g556 g557 g558 g559   
## 224 25 13 46 18 26 77 23 80 65 28 203 96   
## g560 g561 g562 g563 g564 g565 g566 g567 g568 g569 g570 g571 g572   
## 44 227 11 56 66 40 54 94 55 204 198 203 417   
## g573 g574 g575 g576 g577 g578 g579 g580 g581 g582 g583 g584 g585   
## 249 192 377 553 553 190 189 197 28 19 188 8 386   
## g586 g587 g588 g589 g590 g591 g592 g593 g594 g595 g596 g597 g598   
## 197 15 29 93 11 181 57 22 28 105 30 59 14   
## g599 g600 g601 g602 g603 g604 g605 g606 g607 g608 g609 g610 g611   
## 188 3 23 27 2 46 21 19 63 50 18 85 187   
## g612 g613 g614 g615 g616 g617 g618 g619 g620 g621 g622 g623 g624   
## 219 212 210 553 190 209 234 356 198 8 20 32 15   
## g625 g626 g627 g628 g629 g630 g631 g632 g633 g634 g635 g636 g637   
## 16 26 24 10 12 21 67 27 11 12 17 14 25   
## g638 g639 g640 g641 g642 g643 g644 g645 g646 g647 g648 g649 g650   
## 76 12 43 14 11 49 14 16 10 17 26 10 211   
## g651 g652 g653 g654 g655 g656 g657 g658 g659 g660 g661 g662 g663   
## 13 16 75 52 11 7 216 9 26 70 13 12 398   
## g664 g665 g666 g667 g668 g669 g670 g671 g672 g673 g674 g675 g676   
## 8 16 71 11 14 372 8 14 77 22 13 40 12   
## g677 g678 g679 g680 g681 g682 g683 g684 g685 g686 g687 g688 g689   
## 25 50 36 11 59 14 21 17 11 32 19 36 224   
## g690 g691 g692 g693 g694 g695 g696 g697 g698 g699 g700 g701 g702   
## 3 199 6 5 183 10 63 7 32 7 56 43 39   
## g703 g704 g705 g706 g707 g708 g709 g710 g711 g712 g713 g714 g715   
## 32 26 54 20 104 15 56 35 108 34 22 32 69   
## g716 g717 g718 g719 g720 g721 g722 g723 g724 g725 g726 g727 g728   
## 25 29 69 60 23 26 68 24 101 10 58 14 13   
## g729 g730 g731 g732 g733 g734 g735 g736 g737 g738 g739 g740 g741   
## 18 18 41 32 10 25 37 19 23 17 12 31 19   
## g742 g743 g744 g745 g746 g747 g748 g749 g750 g751 g752 g753 g754   
## 11 59 190 33 21 69 66 58 14 553 187 553 204   
## g755 g756 g757 g758 g759 g760 g761 g762 g763 g764 g765 g766 g767   
## 187 210 193 212 208 231 191 188 235 188 215 355 195   
## g768 g769 g770 g771 g772 g773 g774 g775 g776 g777 g778 g779 g780   
## 356 199 187 29 38 41 23 12 72 21 40 34 48   
## g781 g782 g783 g784 g785 g786 g787 g788 g789 g790 g791 g792 g793   
## 191 222 219 227 209 221 223 553 249 195 13 25 15   
## g794 g795 g796 g797 g798 g799 g800 g801 g802 g803 g804 g805 g806   
## 22 63 90 4 17 51 33 242 553 189 385 189 205   
## g807 g808 g809 g810 g811 g812 g813 g814 g815 g816 g817 g818 g819   
## 211 209 235 219 29 89 22 146 20 55 16 80 28   
## g820 g821 g822 g823 g824 g825 g826 g827 g828 g829 g830 g831 g832   
## 74 32 107 18 91 28 22 22 18 38 23 192 229   
## g833 g834 g835 g836 g837 g838 g839 g840 g841 g842 g843 g844 g845   
## 227 186 186 207 191 553 196 200 17 9 57 33 16   
## g846 g847 g848 g849 g850 g851 g852 g853 g854 g855 g856 g857 g858   
## 13 21 20 189 201 388 34 9 386 35 10 11 14   
## g859 g860 g861 g862 g863 g864 g865 g866 g867 g868 g869 g870 g871   
## 17 26 186 223 236 553 228 188 216 191 217 211 36   
## g872 g873 g874 g875 g876 g877 g878 g879 g880 g881 g882 g883 g884   
## 195 27 77 23 32 23 38 13 38 41 57 15 71   
## g885 g886 g887 g888 g889 g890 g891 g892 g893 g894 g895 g896 g897   
## 25 37 53 18 20 38 189 553 553 202 194 225 252   
## g898 g899 g900 g901 g902 g903 g904 g905 g906 g907 g908 g909 g910   
## 393 244 225 11 19 85 35 15 35 35 34 108 191   
## g911 g912 g913 g914 g915 g916 g917 g918 g919 g920 g921 g922 g923   
## 79 30 20 19 52 15 30 35 553 34 48 71 28   
## g924 g925 g926 g927 g928 g929 g930 g931 g932 g933 g934 g935 g936   
## 55 44 192 19 10 30 26 185 376 224 232 192 250   
## g937 g938 g939 g940 g941 g942 g943 g944 g945 g946 g947 g948 g949   
## 217 202 228 199 18 17 22 10 70 32 199 69 36   
## g950 g951 g952 g953 g954 g955 g956 g957 g958 g959 g960 g961 g962   
## 13 187 57 31 12 21 68 26 26 16 19 14 53   
## g963 g964 g965 g966 g967 g968 g969 g970 g971 g972 g973 g974 g975   
## 56 20 83 10 11 14 26 191 553 225 228 222 211   
## g976 g977 g978 g979 g980 g981 g982 g983 g984 g985 g986 g987 g988   
## 227 184 195 205 354 11 50 10 32 188 13 13 14   
## g989 g990 g991 g992 g993 g994 g995 g996 g997 g998 g999 g1000   
## 201 21 13 13 20 29 12 33 18 19 24 17

c <- colSums(is.na(microarray\_data))  
c <- data.frame(c)  
c %>%  
 ggplot(aes(x = c)) +  
 geom\_bar()



**c. Find the genes for which there are more than X% (X=10%, 20%, 50%) missing values.**

c <- colSums(is.na(microarray\_data))  
names(c[c > 0.1 \* ncol(microarray\_data)])

## [1] "g1" "g2" "g12" "g14" "g18" "g21" "g22" "g24" "g25" "g26"   
## [11] "g29" "g39" "g40" "g41" "g46" "g48" "g49" "g51" "g52" "g53"   
## [21] "g54" "g55" "g56" "g57" "g58" "g59" "g60" "g61" "g62" "g63"   
## [31] "g64" "g65" "g66" "g67" "g68" "g69" "g70" "g71" "g72" "g73"   
## [41] "g74" "g75" "g76" "g77" "g78" "g79" "g80" "g81" "g82" "g83"   
## [51] "g84" "g85" "g86" "g87" "g88" "g89" "g90" "g91" "g92" "g93"   
## [61] "g94" "g95" "g96" "g97" "g98" "g99" "g100" "g101" "g102" "g103"  
## [71] "g104" "g105" "g106" "g107" "g108" "g109" "g110" "g111" "g112" "g113"  
## [81] "g114" "g115" "g116" "g117" "g118" "g119" "g120" "g130" "g131" "g132"  
## [91] "g133" "g134" "g135" "g136" "g137" "g138" "g139" "g140" "g142" "g147"  
## [101] "g148" "g151" "g152" "g153" "g154" "g155" "g156" "g157" "g158" "g159"  
## [111] "g160" "g165" "g171" "g172" "g173" "g174" "g175" "g176" "g177" "g178"  
## [121] "g179" "g180" "g196" "g200" "g204" "g210" "g233" "g252" "g260" "g290"  
## [131] "g296" "g297" "g301" "g321" "g329" "g332" "g333" "g334" "g335" "g344"  
## [141] "g351" "g352" "g353" "g354" "g355" "g356" "g357" "g358" "g359" "g360"  
## [151] "g361" "g362" "g363" "g364" "g365" "g366" "g367" "g368" "g369" "g370"  
## [161] "g379" "g381" "g382" "g383" "g384" "g385" "g386" "g387" "g388" "g389"  
## [171] "g390" "g391" "g396" "g400" "g401" "g402" "g403" "g404" "g405" "g406"  
## [181] "g407" "g408" "g409" "g410" "g415" "g417" "g418" "g431" "g432" "g433"  
## [191] "g434" "g435" "g436" "g437" "g438" "g439" "g440" "g445" "g450" "g455"  
## [201] "g461" "g462" "g463" "g464" "g465" "g466" "g467" "g468" "g469" "g470"  
## [211] "g476" "g479" "g491" "g492" "g493" "g494" "g495" "g496" "g497" "g498"  
## [221] "g499" "g500" "g510" "g513" "g515" "g518" "g519" "g520" "g522" "g527"  
## [231] "g531" "g532" "g533" "g534" "g535" "g536" "g537" "g538" "g539" "g540"  
## [241] "g544" "g547" "g558" "g561" "g569" "g570" "g571" "g572" "g573" "g574"  
## [251] "g575" "g576" "g577" "g578" "g579" "g580" "g583" "g585" "g586" "g591"  
## [261] "g595" "g599" "g611" "g612" "g613" "g614" "g615" "g616" "g617" "g618"  
## [271] "g619" "g620" "g650" "g657" "g663" "g669" "g689" "g691" "g694" "g707"  
## [281] "g711" "g724" "g744" "g751" "g752" "g753" "g754" "g755" "g756" "g757"  
## [291] "g758" "g759" "g760" "g761" "g762" "g763" "g764" "g765" "g766" "g767"  
## [301] "g768" "g769" "g770" "g781" "g782" "g783" "g784" "g785" "g786" "g787"  
## [311] "g788" "g789" "g790" "g801" "g802" "g803" "g804" "g805" "g806" "g807"  
## [321] "g808" "g809" "g810" "g814" "g822" "g831" "g832" "g833" "g834" "g835"  
## [331] "g836" "g837" "g838" "g839" "g840" "g849" "g850" "g851" "g854" "g861"  
## [341] "g862" "g863" "g864" "g865" "g866" "g867" "g868" "g869" "g870" "g872"  
## [351] "g891" "g892" "g893" "g894" "g895" "g896" "g897" "g898" "g899" "g900"  
## [361] "g909" "g910" "g919" "g926" "g931" "g932" "g933" "g934" "g935" "g936"  
## [371] "g937" "g938" "g939" "g940" "g947" "g951" "g970" "g971" "g972" "g973"  
## [381] "g974" "g975" "g976" "g977" "g978" "g979" "g980" "g985" "g989"

names(c[c > 0.2 \* ncol(microarray\_data)])

## [1] "g18" "g21" "g29" "g41" "g48" "g51" "g52" "g53" "g55" "g58"   
## [11] "g59" "g60" "g62" "g63" "g64" "g66" "g73" "g74" "g75" "g76"   
## [21] "g77" "g78" "g79" "g82" "g83" "g84" "g88" "g89" "g90" "g91"   
## [31] "g92" "g93" "g94" "g96" "g97" "g98" "g99" "g100" "g101" "g104"  
## [41] "g105" "g106" "g107" "g108" "g109" "g112" "g113" "g114" "g115" "g116"  
## [51] "g118" "g119" "g120" "g131" "g132" "g135" "g137" "g138" "g148" "g153"  
## [61] "g157" "g165" "g171" "g172" "g174" "g176" "g178" "g179" "g204" "g252"  
## [71] "g260" "g290" "g301" "g329" "g351" "g352" "g353" "g355" "g356" "g357"  
## [81] "g358" "g359" "g360" "g362" "g363" "g364" "g367" "g368" "g370" "g379"  
## [91] "g381" "g382" "g383" "g384" "g385" "g386" "g387" "g388" "g389" "g390"  
## [101] "g391" "g396" "g401" "g404" "g405" "g406" "g410" "g417" "g418" "g431"  
## [111] "g432" "g433" "g436" "g437" "g438" "g439" "g440" "g455" "g461" "g462"  
## [121] "g465" "g466" "g469" "g470" "g491" "g493" "g496" "g497" "g498" "g515"  
## [131] "g518" "g519" "g527" "g531" "g532" "g535" "g536" "g537" "g538" "g540"  
## [141] "g547" "g558" "g561" "g569" "g571" "g572" "g573" "g575" "g576" "g577"  
## [151] "g585" "g612" "g613" "g614" "g615" "g617" "g618" "g619" "g650" "g657"  
## [161] "g663" "g669" "g689" "g751" "g753" "g754" "g756" "g758" "g759" "g760"  
## [171] "g763" "g765" "g766" "g768" "g782" "g783" "g784" "g785" "g786" "g787"  
## [181] "g788" "g789" "g801" "g802" "g804" "g806" "g807" "g808" "g809" "g810"  
## [191] "g832" "g833" "g836" "g838" "g850" "g851" "g854" "g862" "g863" "g864"  
## [201] "g865" "g867" "g869" "g870" "g892" "g893" "g894" "g896" "g897" "g898"  
## [211] "g899" "g900" "g919" "g932" "g933" "g934" "g936" "g937" "g938" "g939"  
## [221] "g971" "g972" "g973" "g974" "g975" "g976" "g979" "g980" "g989"

names(c[c > 0.5 \* ncol(microarray\_data)])

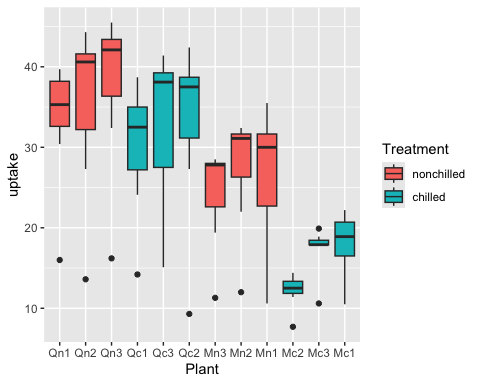
## [1] "g18" "g55" "g58" "g79" "g83" "g99" "g135" "g137" "g138" "g329"  
## [11] "g352" "g368" "g389" "g390" "g417" "g431" "g498" "g519" "g527" "g531"  
## [21] "g538" "g576" "g577" "g615" "g751" "g753" "g788" "g802" "g838" "g864"  
## [31] "g892" "g893" "g919" "g971"

**d. Replace the missing values by the average expression value for the particular gene. (Note: Imputing data has to be used with caution!)**

microarray\_data\_imputed <- data.frame(lapply(microarray\_data, function(x){ifelse(is.na(x), mean(x, na.rm = TRUE), x)}))

**3. Visualize the data in the CO2 dataset in a way that gives you a deeper understanding of the data. What do you see?**

CO2 %>%  
ggplot(aes(x=Plant, y=uptake, fill=Treatment)) +  
 geom\_boxplot()



## That Quebec plants (Qn1-3 and Qc1-3) have a higher uptake and also that nonchilled plants have a slightly higher uptake than chilled plants. ##

***Task 8***

**1. Install the Tidybiology package, which includes the data ‘chromosome’ and ‘proteins’ devtools::install\_github(“hirscheylab/tidybiology”)**

library(tidybiology)  
library(ggpubr)

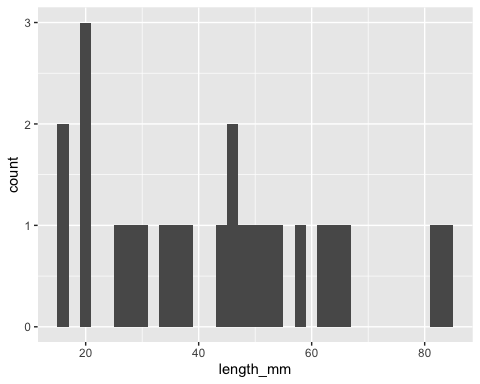
**a. Extract summary statistics (mean, median and maximum) for the following variables from the ‘chromosome’ data: variations, protein coding genes, and miRNAs. Utilize the tidyverse functions to make this as simply as possible.**

chromosome %>%  
 select(protein\_codinggenes, variations, mi\_rna) %>%  
 summary()

## protein\_codinggenes variations mi\_rna   
## Min. : 71.0 Min. : 211643 Min. : 15.00   
## 1st Qu.: 595.8 1st Qu.: 4395298 1st Qu.: 55.75   
## Median : 836.0 Median : 6172346 Median : 75.00   
## Mean : 850.0 Mean : 6484572 Mean : 73.17   
## 3rd Qu.:1055.5 3rd Qu.: 8742592 3rd Qu.: 92.00   
## Max. :2058.0 Max. :12945965 Max. :134.00

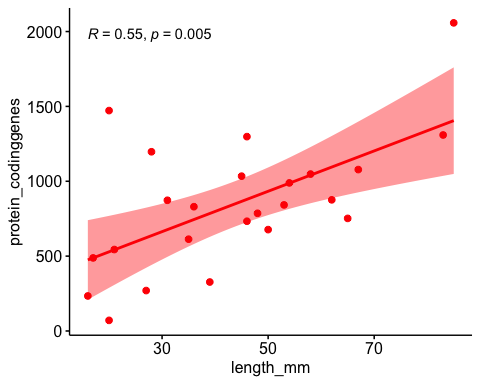
**b. How does the chromosome size distribute? Plot a graph that helps to visualize this by using ggplot2 package functions.**

chromosome %>%  
 ggplot(aes(x=length\_mm)) +  
 geom\_histogram(binwidth=2)

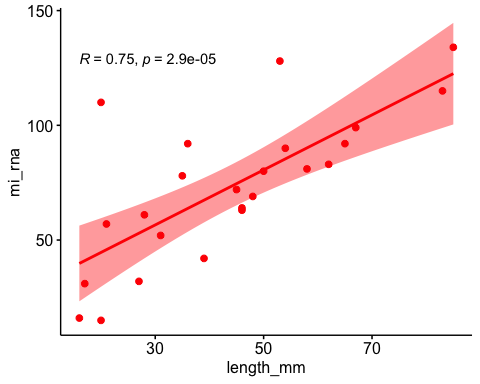


**c. Does the number of protein coding genes or miRNAs correlate with the length of the chromosome? Make two separate plots to visualize these relationships.**

chromosome %>%  
 ggscatter(x = "length\_mm", y = "protein\_codinggenes",   
 add = "reg.line", conf.int = TRUE, color = "red",  
 cor.coef = TRUE, cor.method = "spearman")



chromosome %>%  
 ggscatter(x = "length\_mm", y = "mi\_rna",   
 add = "reg.line", conf.int = TRUE, color = "red",  
 cor.coef = TRUE, cor.method = "spearman")



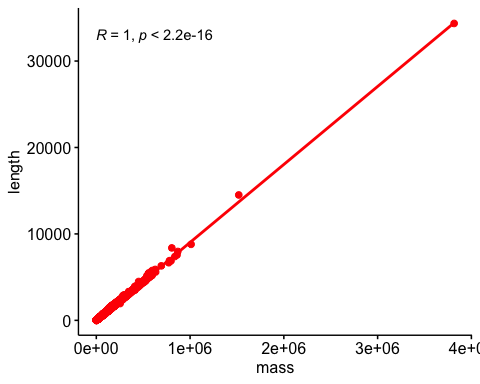
## Both protein coding genes and miRNAs correlate with the length of the chromosome, which would be expected. However miRNAs corrates stronger than protein coding genes ##

**d. Calculate the same summary statistics for the ‘proteins’ data variables length and mass. Create a meaningful visualization of the relationship between these two variables by utilizing the ggplot2 package functions. Play with the colors, theme- and other visualization parameters to create a plot that pleases you.**

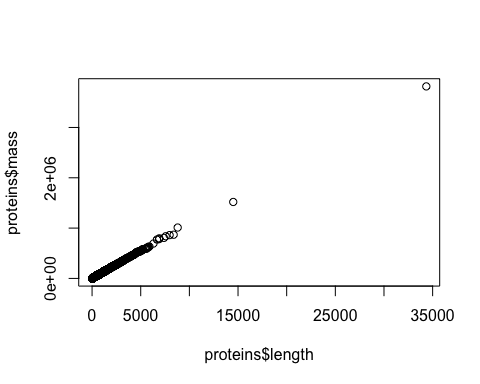
proteins %>%  
 select(length, mass) %>%  
 summary()

## length mass   
## Min. : 2.0 Min. : 260   
## 1st Qu.: 251.0 1st Qu.: 27940   
## Median : 414.0 Median : 46140   
## Mean : 557.2 Mean : 62061   
## 3rd Qu.: 669.0 3rd Qu.: 74755   
## Max. :34350.0 Max. :3816030

proteins %>%  
 ggscatter(x = "mass", y = "length",   
 add = "reg.line", color = "red",  
 cor.coef = TRUE, cor.method = "spearman")



qqplot(proteins$length, proteins$mass)



proteins\_log <- data.frame(lapply(proteins, function(x) if(is.numeric(x)) log(x) else x)) ## Data seems to be skewed.  
  
proteins\_log %>%  
 ggscatter(x = "mass", y = "length",   
 color = "mass",  
 cor.coef = TRUE, cor.method = "pearson", conf.int = TRUE) + #changed to pearson as the data now has normal distribution  
 scale\_color\_gradient(low = "blue", high = "red")

