|  |  |  |
| --- | --- | --- |
| BabyNames %>%  filter(name == "Arjun") %>%  summarise(total = sum(count))  BabyNames %>% nrow() # How many cases in the table  BabyNames %>% names() # The names of the variables  BabyNames %>% head(3) # the first 3 cases  BabyNames %>% str() # view of the first few cases and variable structure  BabyNames %>% tail() # shows the last few cases  # Functions and Explanations  mean()  sum()  n() #find how many cases there are  min()  max()  median()  sd()  n\_distinct()  na.rm = TRUE – calculates even with NA values  age %in% c(18,19,20) or name %in% c("abby", "abe")  ifelse(age > 18, "voter", "non-voter")  filter(year > 1990, sex == "F")  filter(year > 1990 | sex == "F") # pass through either or  filter(year %in% c(1980, 1990, 1991, 1995))  filter(transect %in% c(“1”, “3”)  filter(name == "Prince")  rename(babies = name)  group\_by(year, sex) %>% #groups by the year and sex  summarise(total = sum(count)) %>%  arrange(desc(total))  mtcars %>%  summarise(avg\_count = mean(cyl)) %>%  mutate(ratio = mpg/cyl) %>%  filter(mpg > 30) %>%  select(mpg, cyl, -Year)    if (n == 1) { return(0)} else if (n ==2){ return(1)} {else (n>= 3) {return(fib(n-1) + fib(n-2))} | InfantMortality <- CountryCentroids %>%  select(name, iso\_a3) %>%  left\_join(CountryData %>% select(country, infant), by = c("name" = "country")) # left joins the DF Countrydata matching name to country  MigrationFlows %>%  left\_join(InfantMortality, by = c("destcode" = "iso\_a3"))  BP\_narrow <- BP\_wide %>%  gather(key = when, value = sbp, before, after)  spread(key = when, value = sbp) # splits up the values of when into 2 different columns, their values in their columns  separate(key, into = c("Part", "Measure"), sep = "\\.") sep = "-"  `sapply` \* Takes in list \* Returns vector, matrix, array  `lapply` \* Takes in list \* Returns list  `vapply` \* Takes in list \* Returns vector, matrix, array \* Need to specify FUN.VALUE = numeric(1)  # Graphs  Scorecard\_narrow1 %>% ggplot(aes(x = ADM\_RATE, y = Net\_Cost)) +  geom\_point(shape = 1, color = "blue", alpha = 0.25, aes(color = wday, size = is\_holiday)) + geom\_smooth(color = "red", se = FALSE) + ylim(0,50) + labs(title = "Score Card", x = "Admission rate, %", y = "Net cost per year") + theme\_igray() + theme(plot.title = element\_text(size = 25, color = "black"), axis.title.x = element\_text(size = 15, color = "black"),  axis.title.y = element\_text(size = 15, color = "black")) + facet\_grid (CLASS~Quarter)  facet\_grid – takes 2 variables. Wrap – 1  geom\_smooth(method = lm, method = “loess”) lm – linear. Loess – CI  function(n){x <- rep(NA, n-1) x[1] = 1 x[2] = 2 for (I in 3:(n-1)) x[i] <- x[i-1] + 2  ggplot(aes(x = educ, y = wage, color = sex)) + geom\_point() + geom\_smooth(aes(color = sex), method = "lm") + labs(title = "Wage vs. Education in CPS85") + theme(  plot.title = element\_text(size = 20, color = "black")) + ylim(0,15) | Df[,-c(1,3)] %>% sapply(max) – Applies to all columns but 1 and 3  set.seed(1337)  m <- matrix(runif(15000, -3, 3), ncol = 3)  m.ssq.loop <- rep(0, nrow(m))  **for**(i **in** 1:nrow(m)){  m.ssq.loop[i] <- sum(m[i,]^2)  }  m.ssq.apply <- rep(0,nrow(m))  m.ssq.apply <- apply(m,1,**function**(x) {  **return**(sum(x^2))  })  return(x[x<0])  sum(is.na(x))  length(x[x == “i”])  vec[vec %% 2 == 0] <- 1  vec1 == vec2  To find if any mismatches:  Any((vec1 == vec2) == F)  Mutate(genre = ifelse(t == “1”, “good”, “bad”)  matrix(rep(0,times = 25, nrow = 5))  for (i in 1:5){ for (j in 1:5){if (I == j) {Mat[I,j] <- 1  left\_join() # Output has all the cases from the Left, even if there is no match in the right  right\_join() # Output has only the cases from the Left with a match in the Right  full\_join() # The output will have all the cases from both the Left and the Right  inner\_join() # lets you choose the cases you want in that list  table2 <- table1 %>%  gather(key = Country, value = Value, Algeria, Brazil, Columbia) %>%  select(Country, Year, Value)  group\_by(Year) %>%  mutate(Average = mean(Value, na.rm = TRUE))  table3 <- avg %>%  spread(key = Country, value = Value) %>%  select(Year, Algeria, Brazil, Columbia, Average) |