

Zuo Sqi 2444524 133 MT Extra Credit.

1. Group by orders the data by column & name

Summarise shows two columns the first is primary key, second is the count, size # of names

mutate create a new column showing the rank of each name (descending order)

filter obtain Fernando's info -

2. Group by number instead.

3. month

income is a categorical variable.

peak Oct-Nov Male, CA, MN

bottom Aug-Nov

4. new ← BabyName %>%  
filter(year > 2000)

new ← BabyNames %>%  
spread(key = sex, value = count) %>%  
mutate(allsex = F + M) %>%  
filter(allsex ≥ 20) %>%  
select(year, name)

5. Running time, sex.

quanti qualitative. F/M

not likely the density not the fractions

also no wedge on the data plot.

6. 1. a statistic

22 Runny time	Age (group)	sex
quant	qualit	qualit
	20-30	F
	30-40	M
	...	
	80-90	

Not likely.

7. Besty A CA<sub>m</sub> W Don S

First = Variable 1

8.

Groupby (percent)

summarize (total = n())

arrange (desc (total))

mutate (rank = rank (total))

head()

Groupby (bodytype)

~~summarize~~ (total<sub>m</sub> = n(f), total F = n(f))  
mutate

9.

Temp ← BodyTypes %>%

mutate(totalf = sum(f), totalm = sum(m))

~~select(fratio = sum(f) / totalf~~  
Group by (body-type) %>%

select (fratio = sum(f, na.rm = TRUE) / totalf  
mratio = sum(m, na.rm = TRUE) / totalm)

10.

not == but =

use arrange(desc(total))

after plot %>%

11.

testloop ← function(n)  
  x = vector(type = "num", length = n-1)  
  x[0] = 0

  x[1] = 2

  for (i = 2; i < n; i++)  
  while (i < n) {

    x[i] = x[i-1] + 2

    i++

  }

  return x

tmp %>%

ggplot(aes(x = n, y = test)) + geom\_point() + geom\_smooth()

12.

data %>%

gather(key = genre, value = dNISSG, tuscant, foh\_alum)

Group by (Genre) %>%

~~select~~ select (Genre, sum\_dNISSG = sum(dNISSG))



13. fib(n)

if n == 1  
return(0)

if n == 2  
return(1)

else -

result = fib(n-1) + fib(n-2)

return(result)

14.

Table A % > %

gather (key = Country, value = num, Alg, Drezl, Col) % > %

groupby (Country) % > %

mutate (Avg = mean(num)) % > %

inner\_join (Country) % > %

select (Albhr, Avg)

15.

for i in 1:5 {

for j in 1:5 {

Mat[i][j] = (i == j)

}

}

16.

for i in 1:50

ase  $E_i = \text{ase}[51 - \text{ase}[E_i]]$

for i in 1:50

~~ase~~ mod = ~~i~~ 26

kill = i - 25

~~ase~~  $E_i = \text{ase}[i] + \text{mod} * \text{kill}$

17. `vec = strsplit("um calfor An")`  
`sel = (vec == "i")`  
`sum(sel)`

18. `gather(augquetry, key = type, value = value, Ocean, SolarR, win, Temp)`

19. take only `V1 == a`

i	V1	V2	V3	V4
1	a	1	2	10
2	a	2	8	20

select only `V3` `V4` col

20	10
8	20

transform

Apple	Banana
V2	1
V4	2
V4	10
V4	20

Rename

Apple	Ban
1	1
2	2
10	10
20	20

20 b.  
 for `i` in `1:nrow(data)` {  
`fix(data[i,]) = fix(data[i,])`  
 }

20 c.  
`apply(data, fix)`

21. `diamonds %>%`  
`ggplot(aes(x = Carat, y = price,`  
`color = cut)) +`  
`geom_point(size = 3) +`  
`geom_smooth(lse = false,`  
`color = "blue") +`  
`ylim = (0, 19000)`

22. `belowZero` ~~(X)~~ `<- fun(X)`  
`sel = X < 0`  
`return X[sel]`

20 `fix <- fun(X)`  
`return(X[X == 99] = NA)`

`freezingS = supply(Temp, belowZero)`