

eda

Vrinda

2024-10-10

```
library('ggvis')
library('tidyverse')
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2     3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
## x ggplot2::resolution() masks ggvis::resolution()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library('ggplot2')
library('stats')
library('dplyr')
library('corrplot')
```

```
## corrplot 0.94 loaded
```

```
# read the file
fast_food= read.csv("C:\\Users\\Vrinda\\Downloads\\FastFoodNutritionMenuV3.csv", header=T, na.strings='
# checking the data
head(fast_food)
```

```
##      Company      Item Calories Calories.from.Fat
## 1 McDonald's      Hamburger      250             80
## 2 McDonald's      Cheeseburger      300            110
## 3 McDonald's      Double Cheeseburger      440            210
## 4 McDonald's      McDouble      390            170
## 5 McDonald's      Quarter Pounder® with Cheese      510            230
## 6 McDonald's      Double Quarter Pounder® with Cheese      740            380
##      Total.Fat..g. Saturated.Fat..g. Trans.Fat..g. Cholesterol..mg. Sodium...mg.
## 1          9          3.5          0.5          25          520
## 2         12          6          0.5          40          750
## 3         23         11          1.5          80         1150
## 4         19          8          1          65          920
```

```
## 5          26          12          1.5          90          1190
## 6          42          19          2.5          155          1380
## Carbs..g. Fiber..g. Sugars..g. Protein..g. Weight.Watchers.Pnts
## 1          31           2           6          12          247.5
## 2          33           2           6          15          297
## 3          34           2           7          25          433
## 4          33           2           7          22          383
## 5          40           3           9          29          502
## 6          40           3           9          48          720
```

```
#for evaluating the structure of data
str(fast_food)
```

```
## 'data.frame': 1147 obs. of 14 variables:
## $ Company : chr "McDonald's" "McDonald's" "McDonald's" "McDonald's" ...
## $ Item : chr "Hamburger" "Cheeseburger" "Double Cheeseburger" "McDouble" ...
## $ Calories : chr "250" "300" "440" "390" ...
## $ Calories.from.Fat : chr "80" "110" "210" "170" ...
## $ Total.Fat..g. : chr "9" "12" "23" "19" ...
## $ Saturated.Fat..g. : chr "3.5" "6" "11" "8" ...
## $ Trans.Fat..g. : chr "0.5" "0.5" "1.5" "1" ...
## $ Cholesterol..mg. : chr "25" "40" "80" "65" ...
## $ Sodium..mg. : chr "520" "750" "1150" "920" ...
## $ Carbs..g. : chr "31" "33" "34" "33" ...
## $ Fiber..g. : chr "2" "2" "2" "2" ...
## $ Sugars..g. : chr "6" "6" "7" "7" ...
## $ Protein..g. : chr "12" "15" "25" "22" ...
## $ Weight.Watchers.Pnts: chr "247.5" "297" "433" "383" ...
```

```
# converting string column to numeric
i <- c(3,4,5,6,7,8,9,10,11,12,13,14)
fast_food[, i] <- apply(fast_food[, i], 2, function(x) as.numeric(as.character(x)))
```

```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

```
sapply(fast_food, class)
```

```
##          Company          Item          Calories
##          "character"        "character"        "numeric"
## Calories.from.Fat    Total.Fat..g.    Saturated.Fat..g.
##          "numeric"          "numeric"          "numeric"
```

```
##      Trans.Fat..g.      Cholesterol..mg.      Sodium...mg.
##      "numeric"          "numeric"          "numeric"
##      Carbs..g.          Fiber..g.          Sugars..g.
##      "numeric"          "numeric"          "numeric"
##      Protein..g. Weight.Watchers.Pnts
##      "numeric"          "numeric"
```

```
str(fast_food)
```

```
## 'data.frame':  1147 obs. of  14 variables:
## $ Company      : chr  "McDonald's" "McDonald's" "McDonald's" "McDonald's" ...
## $ Item          : chr  "Hamburger" "Cheeseburger" "Double Cheeseburger" "McDouble" ...
## $ Calories      : num  250 300 440 390 510 740 540 460 510 790 ...
## $ Calories.from.Fat : num  80 110 210 170 230 380 260 220 250 350 ...
## $ Total.Fat..g.  : num  9 12 23 19 26 42 29 24 28 39 ...
## $ Saturated.Fat..g. : num  3.5 6 11 8 12 19 10 8 11 17 ...
## $ Trans.Fat..g.  : num  0.5 0.5 1.5 1 1.5 2.5 1.5 1.5 1.5 2 ...
## $ Cholesterol..mg. : num  25 40 80 65 90 155 75 70 85 145 ...
## $ Sodium...mg.   : num  520 750 1150 920 1190 1380 1040 720 960 2070 ...
## $ Carbs..g.      : num  31 33 34 33 40 40 45 37 38 63 ...
## $ Fiber..g.      : num  2 2 2 2 3 3 3 3 3 4 ...
## $ Sugars..g.     : num  6 6 7 7 9 9 9 8 8 13 ...
## $ Protein..g.    : num  12 15 25 22 29 48 25 24 27 45 ...
## $ Weight.Watchers.Pnts: num  248 297 433 383 502 ...
```

```
#summary of dataset
summary(fast_food)
```

```
##      Company      Item      Calories      Calories.from.Fat
## Length:1147      Length:1147      Min.   : 0.0      Min.   : 0.0
## Class :character  Class :character  1st Qu.: 140.0    1st Qu.: 0.0
## Mode  :character  Mode  :character  Median : 240.0    Median : 80.0
##                                     Mean   : 287.4      Mean   :117.6
##                                     3rd Qu.: 390.0    3rd Qu.:170.0
##                                     Max.    :1220.0    Max.    :750.0
##                                     NA's    :14        NA's    :517
## Total.Fat..g.      Saturated.Fat..g. Trans.Fat..g.      Cholesterol..mg.
## Min.   : 0.00      Min.   : 0.000      Min.   :0.0000      Min.   : 0.00
## 1st Qu.: 0.00      1st Qu.: 0.000      1st Qu.:0.0000      1st Qu.: 0.00
## Median : 8.00      Median : 3.000      Median :0.0000      Median : 20.00
## Mean   :11.68      Mean   : 4.068      Mean   :0.1409      Mean   : 40.65
## 3rd Qu.:18.00      3rd Qu.: 6.000      3rd Qu.:0.0000      3rd Qu.: 50.00
## Max.    :98.00      Max.    :33.000      Max.    :4.5000      Max.    :575.00
## NA's    :68        NA's    :68        NA's    :68        NA's    :28
## Sodium...mg.      Carbs..g.      Fiber..g.      Sugars..g.
## Min.   : 0.0      Min.   : 0.00      Min.   : 0.000      Min.   : 0.00
## 1st Qu.: 70.0      1st Qu.: 17.00      1st Qu.: 0.000      1st Qu.: 2.00
## Median : 190.0      Median : 34.00      Median : 0.000      Median : 8.00
## Mean   : 427.7      Mean   : 38.95      Mean   : 1.455      Mean   : 24.07
## 3rd Qu.: 680.0      3rd Qu.: 52.00      3rd Qu.: 2.000      3rd Qu.: 40.00
## Max.    :2890.0      Max.    :270.00      Max.    :31.000      Max.    :264.00
## NA's    :15        NA's    :69        NA's    :83        NA's    :29
## Protein..g.      Weight.Watchers.Pnts
```

```
## Min.      : 0.000    Min.      : 0.0
## 1st Qu.: 0.000    1st Qu.: 142.8
## Median : 7.000    Median : 272.0
## Mean    : 9.435    Mean    : 310.3
## 3rd Qu.:14.000    3rd Qu.: 430.0
## Max.     :71.000    Max.     :1317.0
## NA's     :68       NA's      :271
```

*#assigning string to factors*

```
fast_food$Company = as.factor(fast_food$Company)
fast_food$Item = as.factor(fast_food$Item)

str(fast_food)
```

```
## 'data.frame':    1147 obs. of  14 variables:
## $ Company      : Factor w/ 6 levels "Burger King",...: 3 3 3 3 3 3 3 3 3 ...
## $ Item         : Factor w/ 1071 levels " Quarter Pounder® with Cheese",...: 419 184 339 659 1
## $ Calories     : num  250 300 440 390 510 740 540 460 510 790 ...
## $ Calories.from.Fat : num  80 110 210 170 230 380 260 220 250 350 ...
## $ Total.Fat..g.  : num  9 12 23 19 26 42 29 24 28 39 ...
## $ Saturated.Fat..g. : num  3.5 6 11 8 12 19 10 8 11 17 ...
## $ Trans.Fat..g.  : num  0.5 0.5 1.5 1 1.5 2.5 1.5 1.5 1.5 2 ...
## $ Cholesterol..mg. : num  25 40 80 65 90 155 75 70 85 145 ...
## $ Sodium...mg.   : num  520 750 1150 920 1190 1380 1040 720 960 2070 ...
## $ Carbs..g.      : num  31 33 34 33 40 40 45 37 38 63 ...
## $ Fiber..g.      : num  2 2 2 2 3 3 3 3 3 4 ...
## $ Sugars..g.     : num  6 6 7 7 9 9 9 8 8 13 ...
## $ Protein..g.    : num  12 15 25 22 29 48 25 24 27 45 ...
## $ Weight.Watchers.Pnts: num  248 297 433 383 502 ...
```

*#checking null values*

```
colSums(is.na(fast_food))
```

```
##           Company           Item           Calories
##           0              0              14
## Calories.from.Fat    Total.Fat..g.    Saturated.Fat..g.
##           517             68             68
## Trans.Fat..g.      Cholesterol..mg.    Sodium...mg.
##           68             28             15
## Carbs..g.          Fiber..g.          Sugars..g.
##           69             83             29
## Protein..g. Weight.Watchers.Pnts
##           68             271
```

*#dropping calories.from.fat column*

```
df = subset(fast_food,select = -c(Calories.from.Fat))

head(df)
```

```
##           Company           Item    Calories    Total.Fat..g.
## 1 McDonald's      Hamburger      250           9
## 2 McDonald's      Cheeseburger    300          12
```

```
## 3 McDonald's          Double Cheeseburger      440          23
## 4 McDonald's          McDouble                390          19
## 5 McDonald's          Quarter Pounder® with Cheese 510          26
## 6 McDonald's Double Quarter Pounder® with Cheese 740          42
##   Saturated.Fat..g. Trans.Fat..g. Cholesterol..mg. Sodium...mg. Carbs..g.
## 1           3.5         0.5           25           520          31
## 2           6.0         0.5           40           750          33
## 3          11.0         1.5           80          1150          34
## 4           8.0         1.0           65           920          33
## 5          12.0         1.5           90          1190          40
## 6          19.0         2.5          155          1380          40
##   Fiber..g. Sugars..g. Protein..g. Weight.Watchers.Pnts
## 1           2           6           12           247.5
## 2           2           6           15           297.0
## 3           2           7           25           433.0
## 4           2           7           22           383.0
## 5           3           9           29           502.0
## 6           3           9           48           720.0
```

```
#removing na values
#df = na.omit(df)

head(df)
```

```
##      Company          Item Calories Total.Fat..g.
## 1 McDonald's          Hamburger      250          9
## 2 McDonald's          Cheeseburger     300         12
## 3 McDonald's          Double Cheeseburger 440         23
## 4 McDonald's          McDouble        390         19
## 5 McDonald's          Quarter Pounder® with Cheese 510         26
## 6 McDonald's Double Quarter Pounder® with Cheese 740         42
##   Saturated.Fat..g. Trans.Fat..g. Cholesterol..mg. Sodium...mg. Carbs..g.
## 1           3.5         0.5           25           520          31
## 2           6.0         0.5           40           750          33
## 3          11.0         1.5           80          1150          34
## 4           8.0         1.0           65           920          33
## 5          12.0         1.5           90          1190          40
## 6          19.0         2.5          155          1380          40
##   Fiber..g. Sugars..g. Protein..g. Weight.Watchers.Pnts
## 1           2           6           12           247.5
## 2           2           6           15           297.0
## 3           2           7           25           433.0
## 4           2           7           22           383.0
## 5           3           9           29           502.0
## 6           3           9           48           720.0
```

```
#checking correlation between columns
correlation_table = cor(df[,3:13],use = 'complete.obs')
print(correlation_table)
```

```
##      Calories Total.Fat..g. Saturated.Fat..g. Trans.Fat..g.
## Calories      1.0000000    0.8240320    0.80432130    0.57871050
## Total.Fat..g.    0.8240320    1.0000000    0.88821190    0.57535310
```

## Saturated.Fat..g.	0.8043213	0.8882119	1.00000000	0.70076258
## Trans.Fat..g.	0.5787105	0.5753531	0.70076258	1.00000000
## Cholesterol..mg.	0.6252053	0.7210375	0.72628753	0.38015473
## Sodium...mg.	0.7179522	0.8253916	0.67517285	0.35821025
## Carbs..g.	0.6583407	0.1366090	0.23010921	0.20484939
## Fiber..g.	0.4449910	0.4455613	0.31132008	0.11634982
## Sugars..g.	0.2984883	-0.2352715	-0.07454486	0.04967015
## Protein..g.	0.7442321	0.8173856	0.76749037	0.63587200
## Weight.Watchers.Pnts	0.9884115	0.7416688	0.75019460	0.54972549
##	Cholesterol..mg.	Sodium...mg.	Carbs..g.	Fiber..g.
## Calories	0.6252053	0.7179522	0.6583407	0.4449910
## Total.Fat..g.	0.7210375	0.8253916	0.1366090	0.4455613
## Saturated.Fat..g.	0.7262875	0.6751729	0.2301092	0.3113201
## Trans.Fat..g.	0.3801547	0.3582103	0.2048494	0.1163498
## Cholesterol..mg.	1.0000000	0.6856512	0.1135675	0.2702619
## Sodium...mg.	0.6856512	1.0000000	0.1306725	0.5881992
## Carbs..g.	0.1135675	0.1306725	1.0000000	0.1615138
## Fiber..g.	0.2702619	0.5881992	0.1615138	1.0000000
## Sugars..g.	-0.1692814	-0.2852356	0.8679442	-0.2053926
## Protein..g.	0.6980781	0.8102137	0.1157212	0.4460572
## Weight.Watchers.Pnts	0.5613171	0.6278657	0.7570152	0.3855712
##	Sugars..g.	Protein..g.	Weight.Watchers.Pnts	
## Calories	0.29848834	0.7442321	0.9884115	
## Total.Fat..g.	-0.23527150	0.8173856	0.7416688	
## Saturated.Fat..g.	-0.07454486	0.7674904	0.7501946	
## Trans.Fat..g.	0.04967015	0.6358720	0.5497255	
## Cholesterol..mg.	-0.16928142	0.6980781	0.5613171	
## Sodium...mg.	-0.28523561	0.8102137	0.6278657	
## Carbs..g.	0.86794418	0.1157212	0.7570152	
## Fiber..g.	-0.20539255	0.4460572	0.3855712	
## Sugars..g.	1.00000000	-0.2143395	0.4377990	
## Protein..g.	-0.21433945	1.0000000	0.6559830	
## Weight.Watchers.Pnts	0.43779904	0.6559830	1.0000000	

**UNSATURATED FAT:** These are the good fats, and doctors say they should be the majority of fat that people eat. For cooking, they usually come in the form of liquid oils, not solid fats. Unsaturated fats are listed on food packages as polyunsaturated fats and monounsaturated fats.

**SATURATED FAT:** These fats are often derived from animals and generally take a more solid form. They raise “bad” cholesterol and can contribute to heart disease. The government recommends that saturated fats make up less than 10 per cent of daily calories. Common sources include:

- high-fat cheeses
- high-fat cuts of meat
- whole-fat milk and cream and ice cream
- butter
- palm and coconut oils

**TRANS FAT:** These are the worst fats, and the FDA is forcing food companies to phase them out. They are made when hydrogen is added to vegetable oil, usually to create a certain consistency or increase shelf life, and they are also called partially hydrogenated oils. Many of them have already been phased out, but foods that are more likely to contain trans fats are:

- fried items

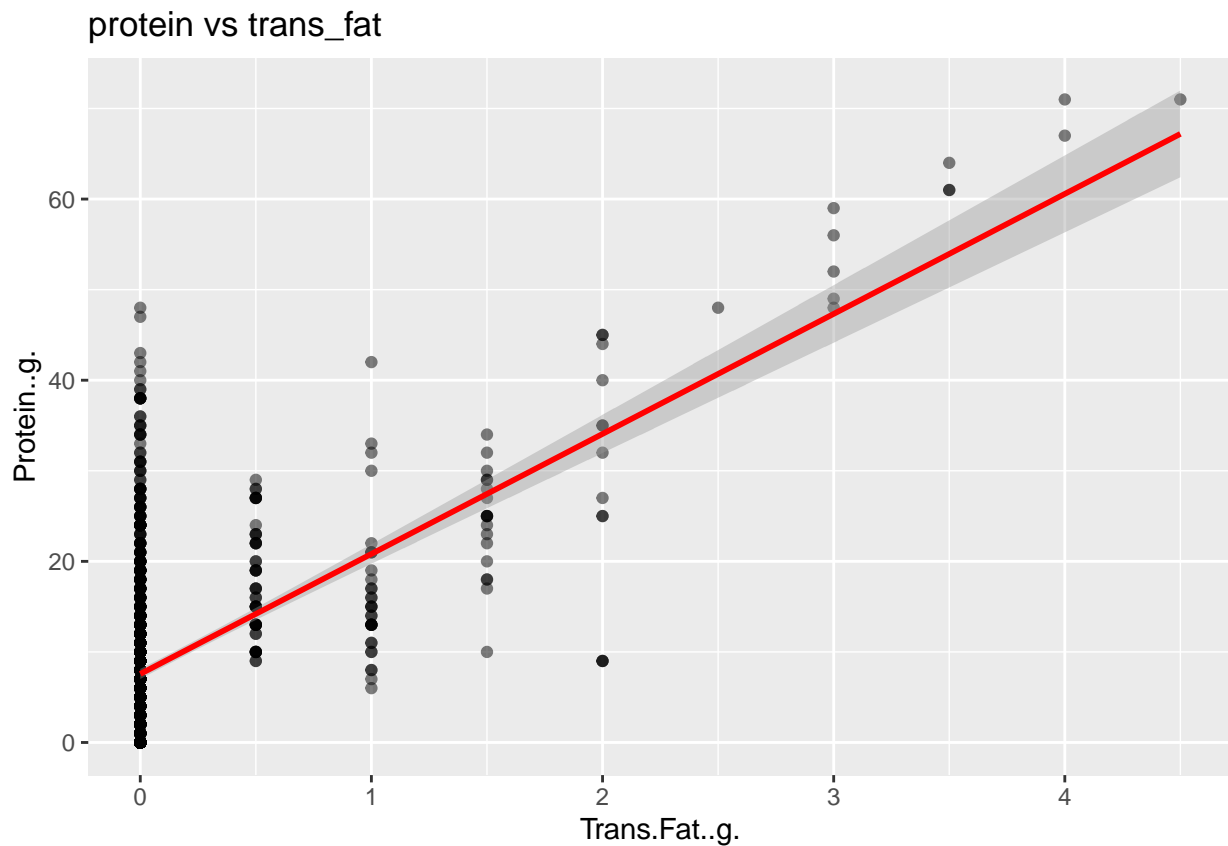
- pie crusts
- stick margarine
- ready-to-use frosting
- coffee creamers
- some microwave popcorn and frozen pizza some cakes, crackers and cookies

```
#check few column relationship through graph
ggplot(df,aes(x = Trans.Fat..g.,y = Protein..g. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',se
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 68 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 68 rows containing missing values or values outside the scale range
## ('geom_point()').
```



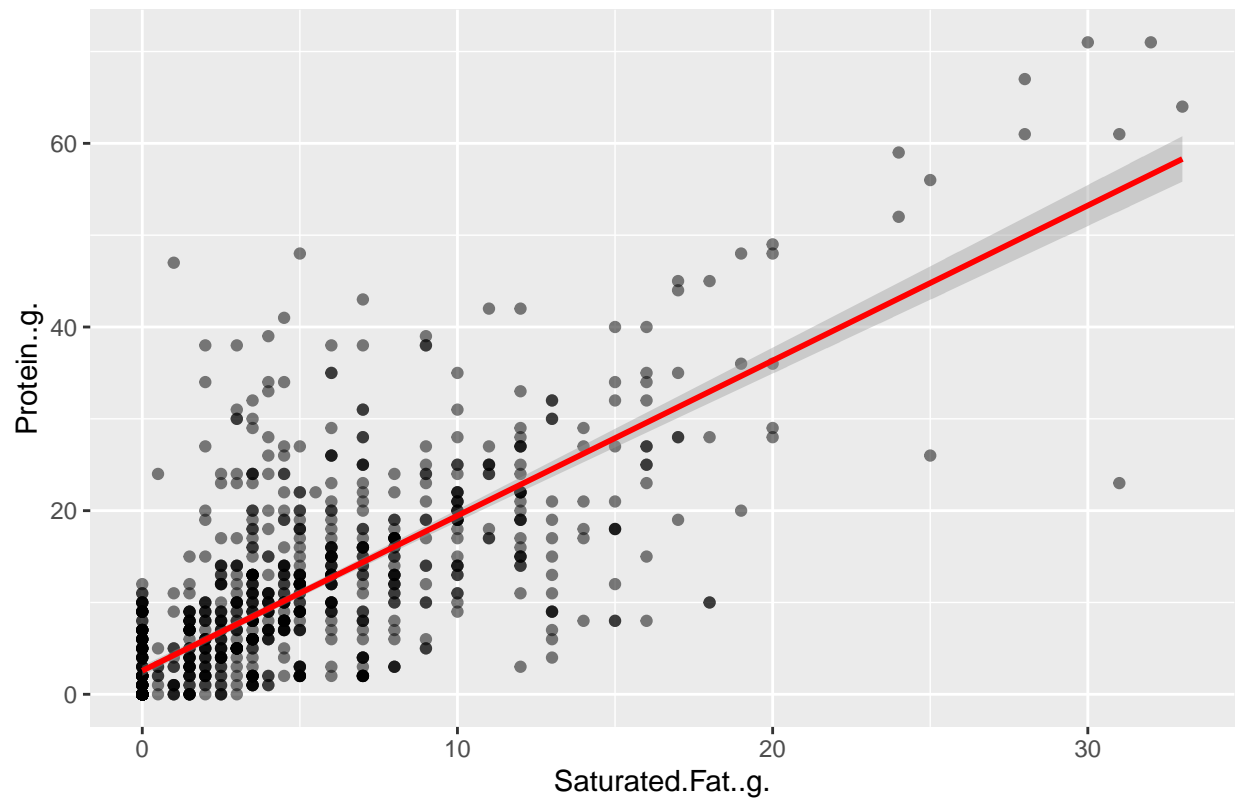
```
ggplot(df,aes(x = Saturated.Fat..g.,y = Protein..g. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',se
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 68 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Removed 68 rows containing missing values or values outside the scale range
## ('geom_point()').
```

protein vs saturated\_fat



```
ggplot(df,aes(x = Calories,y = Protein..g. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',se = TRUE)
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 68 rows containing non-finite outside the scale range
```

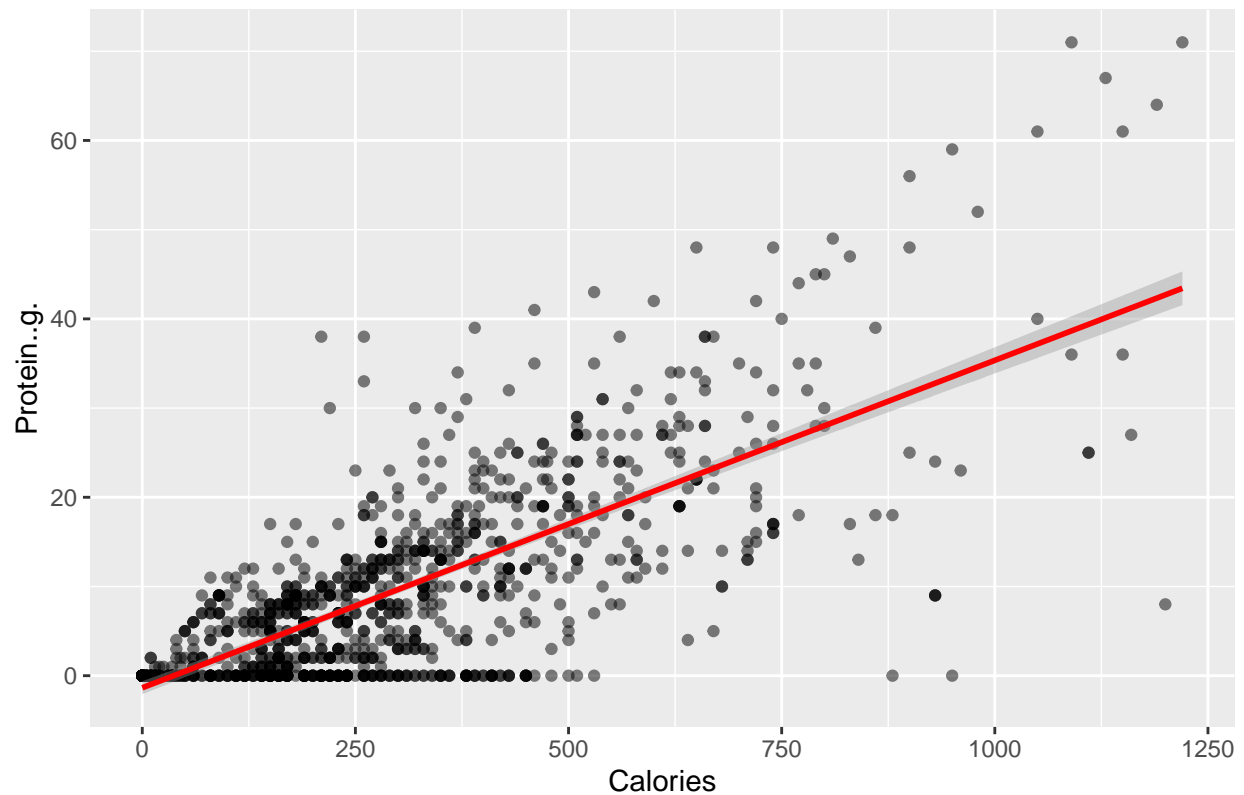
```
## ('stat_smooth()').
```

```
## Removed 68 rows containing missing values or values outside the scale range
```

```
## ('geom_point()').
```



protein vs calories



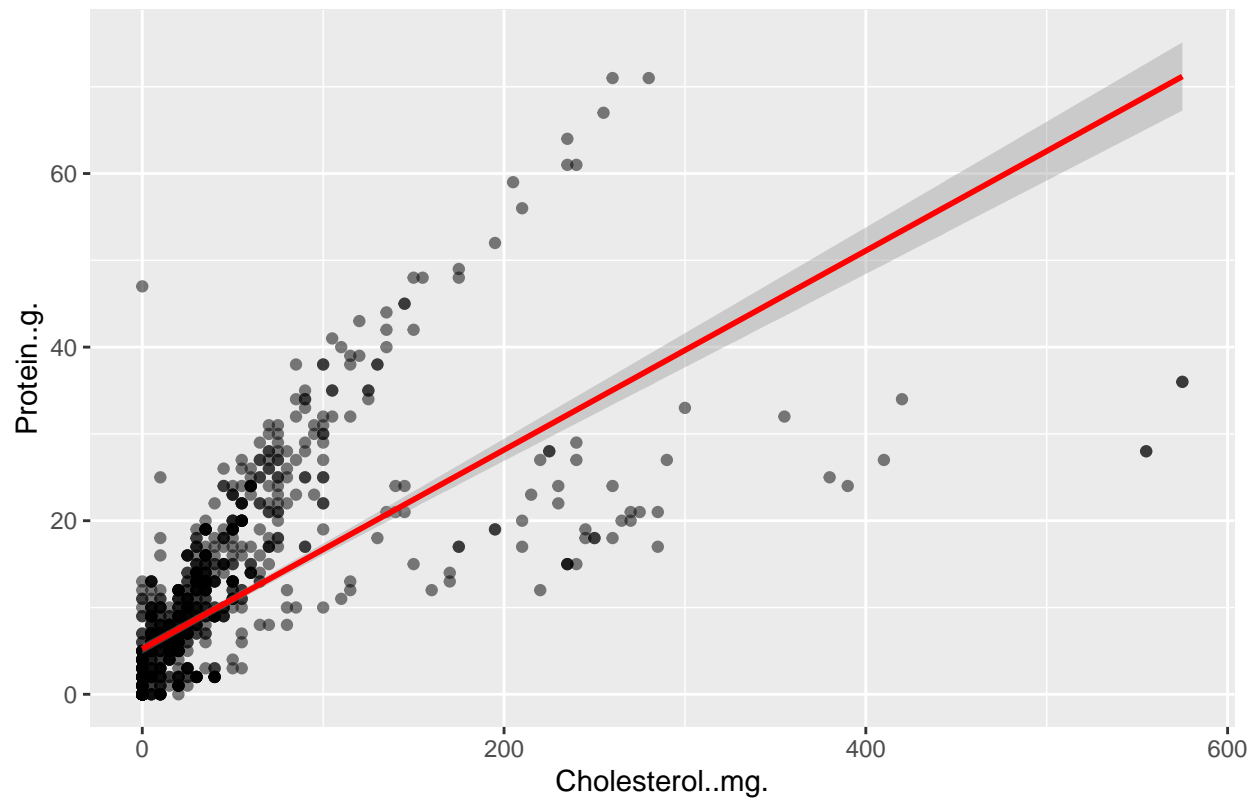
```
ggplot(df,aes(x = Cholesterol..mg.,y = Protein..g. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm'
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 82 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 82 rows containing missing values or values outside the scale range
## ('geom_point()').
```

protein vs cholesterol

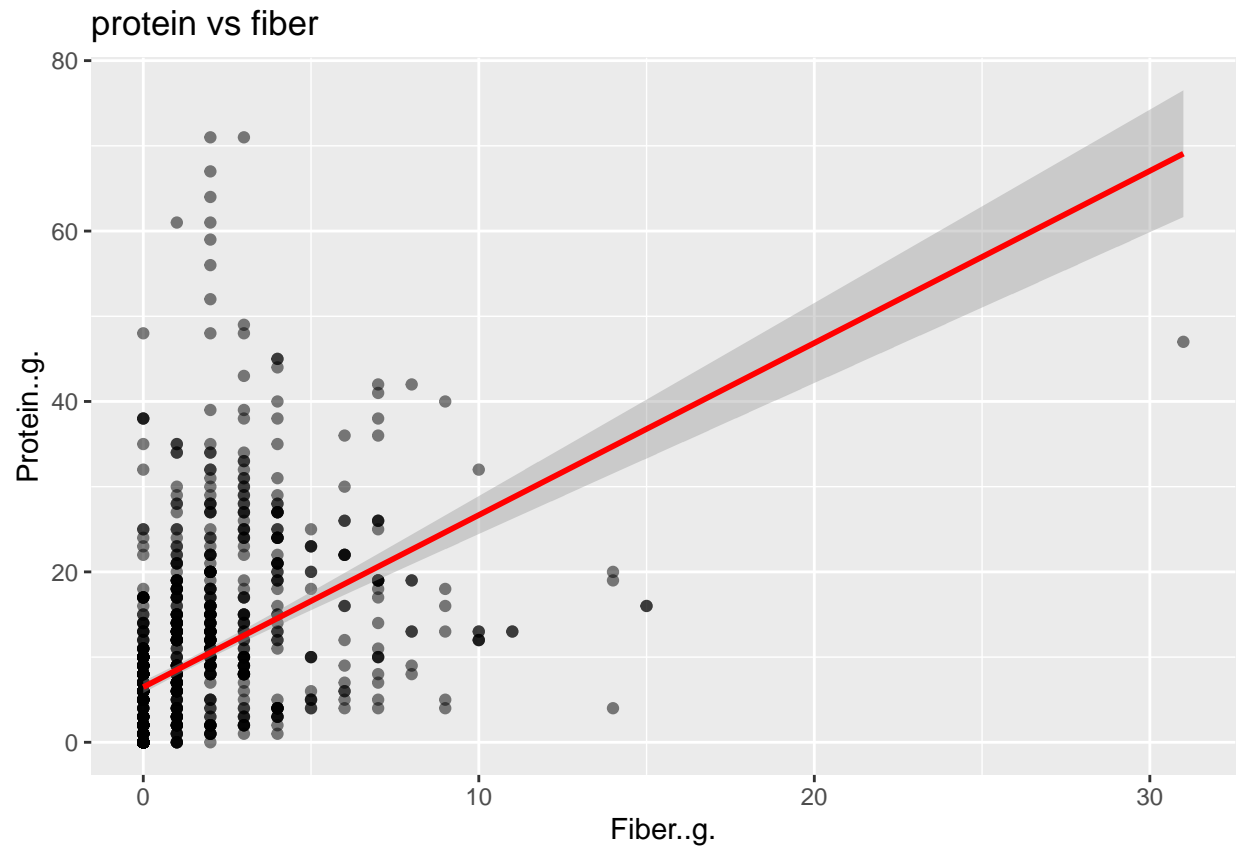


```
ggplot(df,aes(x = Fiber..g.,y = Protein..g. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',se = 'conf')
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 83 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 83 rows containing missing values or values outside the scale range
## ('geom_point()').
```

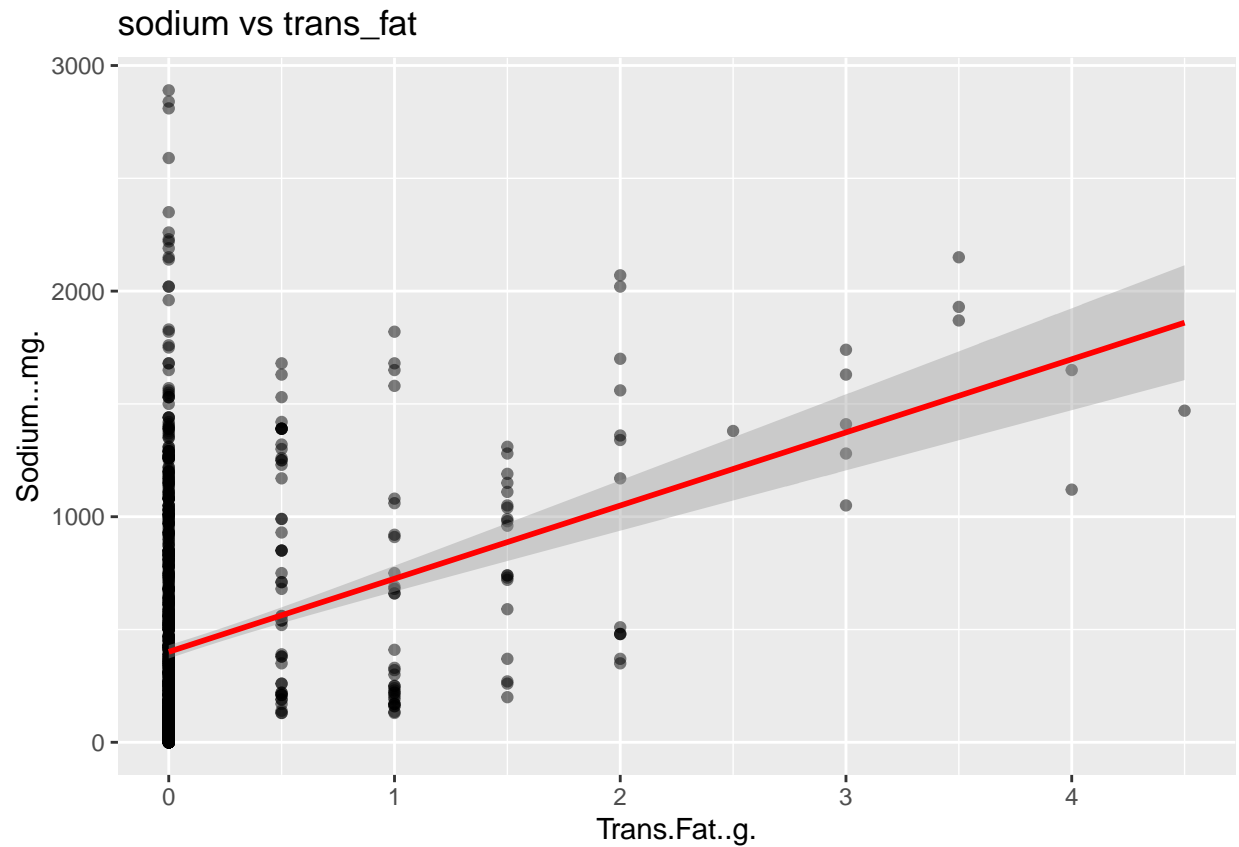


```
ggplot(df,aes(x = Trans.Fat..g.,y = Sodium...mg. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',s
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 68 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 68 rows containing missing values or values outside the scale range
## ('geom_point()').
```



```
ggplot(df,aes(x = Saturated.Fat..g.,y = Sodium...mg. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm')
```

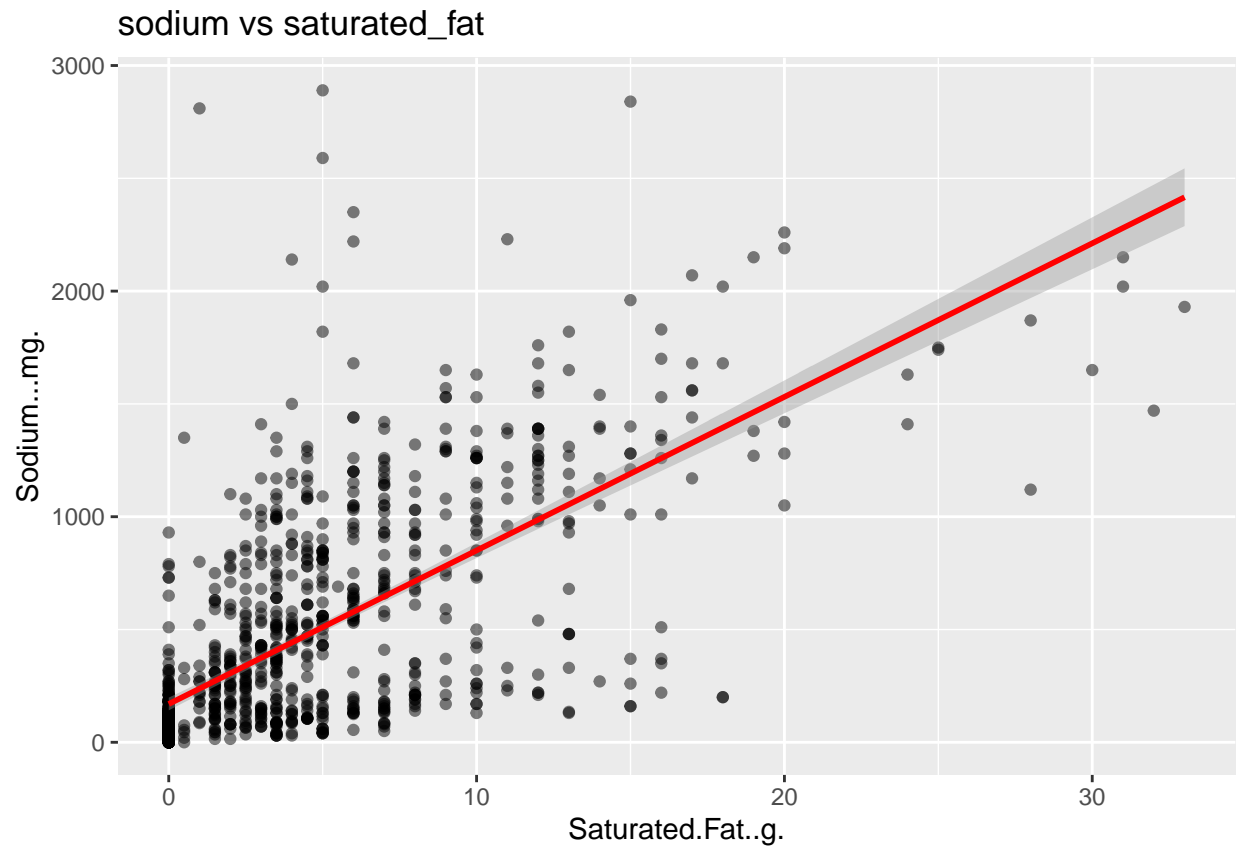
```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 68 rows containing non-finite outside the scale range
```

```
## ('stat_smooth()').
```

```
## Removed 68 rows containing missing values or values outside the scale range
```

```
## ('geom_point()').
```

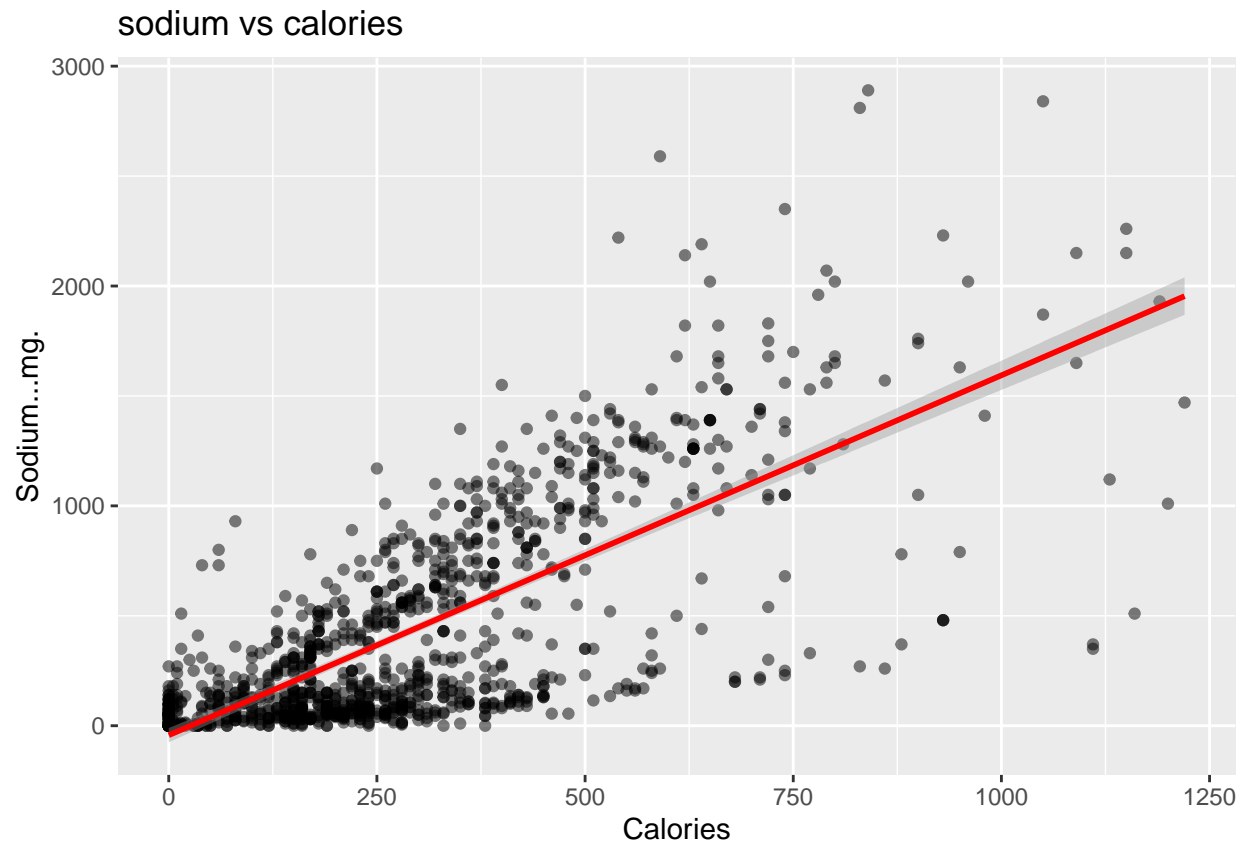


```
ggplot(df,aes(x = Calories,y = Sodium...mg. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',se = TRUE)
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 15 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## ('geom_point()').
```

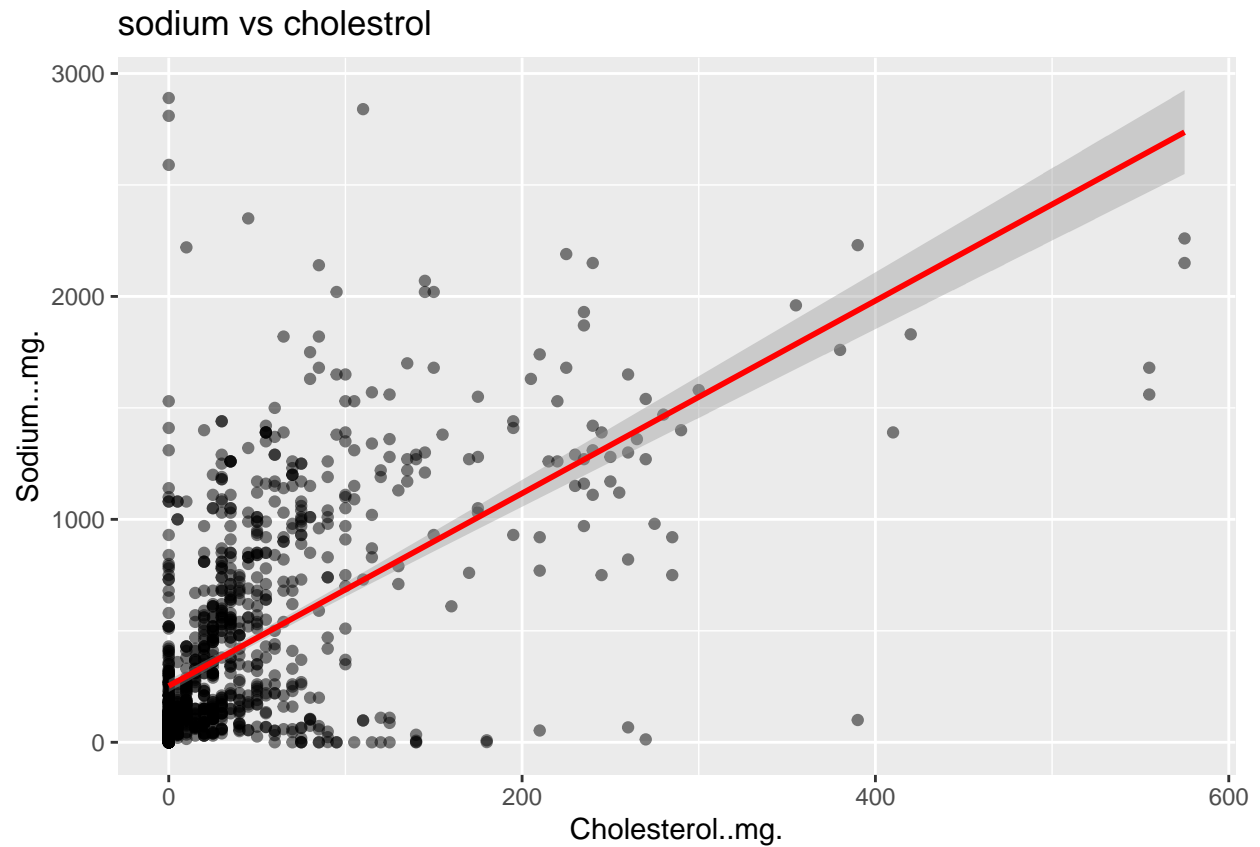


```
ggplot(df,aes(x = Cholesterol..mg.,y = Sodium...mg. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 29 rows containing non-finite outside the scale range  
## ('stat_smooth()').
```

```
## Warning: Removed 29 rows containing missing values or values outside the scale range  
## ('geom_point()').
```



```
ggplot(df,aes(x = Fiber..g.,y = Sodium...mg. ))+geom_point(alpha = 0.5)+geom_smooth(method = 'lm',se =
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 83 rows containing non-finite outside the scale range  
## ('stat_smooth()').
```

```
## Warning: Removed 83 rows containing missing values or values outside the scale range  
## ('geom_point()').
```

sodium vs fiber

