Summarizing data

Start with HTRU2

```
# Read in the CSV
HTRU2 <-
  read_csv(here("Data_Analyses_MATH_208/Datasets/HTRU2/HTRU_2.csv"),
           col_names=FALSE)
# Name the variables
names(HTRU2) = c("Mean_IP", "SD_IP", "EK_IP", "SKW_IP",
                 "Mean_DMSNR", "SD_DMSNR", "EK_DMSNR", "SKW_DMSNR",
                 "Class")
```

```
HTRU2 <- HTRU2 %>%
  mutate(Class=factor(ifelse(Class==0,"Negative","Positive")))
```

condition yes no

Overall (or marginal) numerical summaries

```
HTRU2 %>% summarise(Avg = mean(Mean_IP),
                    Med = median(Mean_IP),
                    '25%ile' = quantile(Mean_IP,0.25),
                    '75%ile' = quantile(Mean_IP,0.25),
                    StD = sd(Mean_IP),
                    IQR = IQR(Mean_IP)
                    )
```

```
# A tibble: 1 x 6
    Avg   Med `25%ile` `75%ile`   StD   IQR
  <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl>
1  111.  115.     101.     101.  25.7  26.2
```

Numerical summaries by group

```
HTRU2 %>% group_by(Class) %>%
  summarise(Avg = mean(Mean_IP),
            Med = median(Mean_IP),
            Q25 = quantile(Mean_IP,0.25),
            Q75 = quantile(Mean_IP,0.75),
            StD = sd(Mean_IP),
            IQR = IQR(Mean_IP))
```

```
# A tibble: 2 x 7
  Class      Avg   Med   Q25   Q75   StD   IQR
  <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Negative 117.  117.  105.  128.   17.5  23.0
2 Positive  56.7  54.3  31.8  79.3  30.0  47.5
```

Summarizing more than one variable at a time

group column

```
HTRU2 %>% group_by(Class) %>% select(Class,Mean_IP,Mean_DMSNR) %>%
  summarise_all(list(Avg=mean,Med=median))
```

```
# A tibble: 2 x 5
  Class    Mean_IP_Avg Mean_DMSNR_Avg Mean_IP_Med Mean_DMSNR_Med
  <fct>          <dbl>          <dbl>       <dbl>          <dbl>
1 Negative       117.            8.86       117.            2.64
2 Positive        56.7          49.8         54.3          33.5
```

In a tibble, everything in the same column should have the same mode

Summarizing more than one variable at a time (longer)

stay the same

```
HTRU2 %>% group_by(Class) %>% select(Class,Mean_IP,Mean_DMSNR) %>%
  summarise_all(list(Avg=mean,Med=median)) %>%
  pivot_longer(cols=starts_with("Mean"),names_to = "Measure") %>%
  arrange(desc(Measure))
```

c(Mean_IP_Med, Mean_IP_Avg, Mean_DMSNR_Med, Mean_DMSNR_Avg)

```
# A tibble: 8 x 3
  Class    Measure          value
  <fct>    <chr>            <dbl>
1 Negative Mean_IP_Med     117.
2 Positive Mean_IP_Med      54.3
3 Negative Mean_IP_Avg     117.
4 Positive Mean_IP_Avg      56.7
5 Negative Mean_DMSNR_Med    2.64
6 Positive Mean_DMSNR_Med   33.5
7 Negative Mean_DMSNR_Avg    8.86
8 Positive Mean_DMSNR_Avg   49.8
```

Summarizing more than one variable at a time (wider)

```
HTRU2 %>% group_by(Class) %>% select(Class,Mean_IP,Mean_DMSNR) %>%
  summarise_all(list(Avg=mean,Med=median,
      Q25 =~quantile(.,probs=c(0.25)),
      Q75 =~quantile(.,probs=c(0.75)))) %>%
  pivot_longer(cols=starts_with("Mean"),names_to = "Measure") %>%
  pivot_wider(id_cols=Measure,names_from=Class) %>%
  arrange(desc(Measure))
```

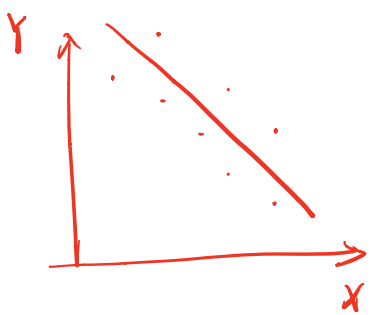
call a func with additional args

unique row identifier

```
# A tibble: 8 x 3
  Measure        Negative Positive
  <chr>             <dbl>    <dbl>
1 Mean_IP_Q75       128.      79.3
2 Mean_IP_Q25       105.      31.8
3 Mean_IP_Med       117.      54.3
4 Mean_IP_Avg       117.      56.7
5 Mean_DMSNR_Q75      4.23    78.3
6 Mean_DMSNR_Q25      1.86    12.8
7 Mean_DMSNR_Med      2.64    33.5
8 Mean_DMSNR_Avg      8.86    49.8
```

Correlation

R^2 = % variance explained in Y by a line in X



Summaries involving two variables

```
HTRU2 %>% group_by(Class) %>%
  summarise(Cor_MeanIP_Mean_DMSNR =
                      cor(Mean_IP,Mean_DMSNR))
```

```
# A tibble: 2 x 2
  Class    Cor_MeanIP_Mean_DMSNR
  <fct>                    <dbl>
1 Negative                 0.117
2 Positive                -0.542
```

Summaries involving two variables

```
ggplot(HTRU2,aes(x=Mean_IP,y=Mean_DMSNR,col=Class)) +
  geom_point() + facet_wrap(~Class) +
  labs(x="Mean IP", y="Mean DMSNR",
       title="Mean IP vs. Mean DMSNR") +
  theme(legend.position = "none") +
  geom_smooth(method="lm",col="black")
```

Summaries involving two variables

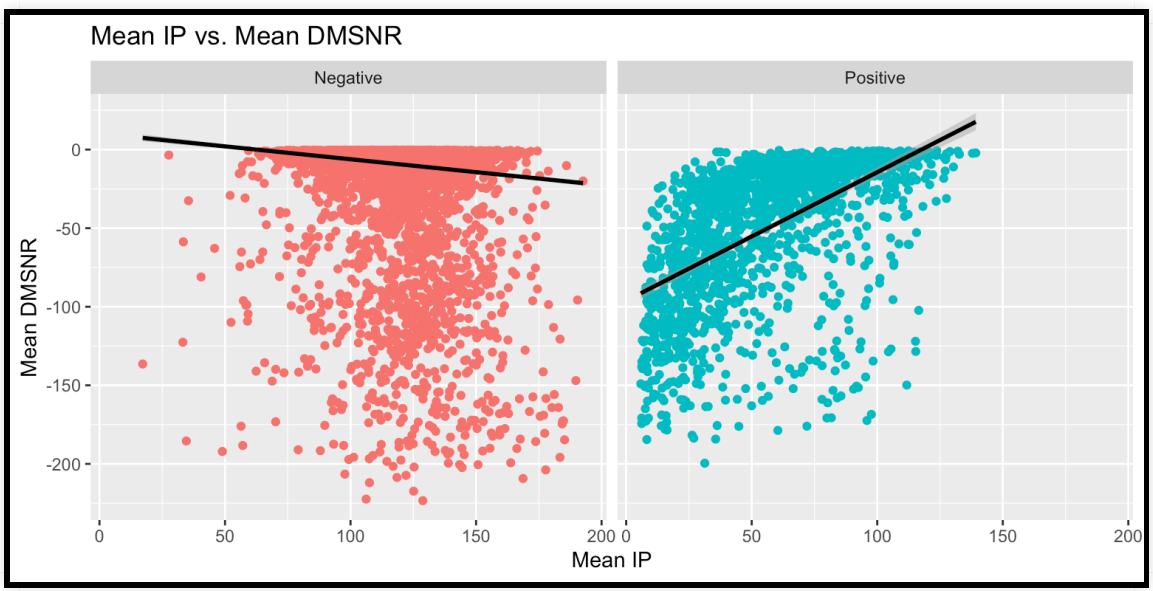
```
HTRU2 <- HTRU2 %>% mutate(Neg_MDMSNR=-Mean_DMSNR)
HTRU2 %>% group_by(Class) %>% summarise(Cor2 =
                                          cor(Mean_IP,Neg_MDMSNR))
```

```
# A tibble: 2 x 2
  Class      Cor2
  <fct>     <dbl>
1 Negative -0.117
2 Positive  0.542
```

opposite

Summaries involving two variables

```
ggplot(HTRU2,aes(x=Mean_IP,y=Neg_MDMSNR,col=Class)) +
  geom_point() + facet_wrap(~Class) +
  labs(x="Mean IP", y="Mean DMSNR",
       title="Mean IP vs. Mean DMSNR") +
  theme(legend.position = "none") +
  geom_smooth(method="lm",col="black")
```



Qualitative data and factors

Boston Crime Data Set

```
crime <- read_csv(
  here("Data_Analyses_MATH_208/Datasets/BostonCrime/crime.csv"))
```

```
head(crime)
```

```
# A tibble: 6 x 17
  INCIDENT_NUMBER OFFENSE_CODE OFFENSE_CODE_GR… OFFENSE_DESCRIP… DISTRIC
  <chr>                  <dbl> <chr>            <chr>            <chr>
1 I182080058              2403 Disorderly Cond… DISTURBING THE … E18
2 I182080053              3201 Property Lost    PROPERTY - LOST  D14
3 I182080052              2647 Other            THREATS TO DO B… B2
4 I182080051               413 Aggravated Assa… ASSAULT - AGGRA… A1
5 I182080050              3122 Aircraft         AIRCRAFT INCIDE… A7
6 I182080049              1402 Vandalism        VANDALISM        C11
# … with 12 more variables: REPORTING_AREA <dbl>, SHOOTING <lgl>,
#   OCCURRED_ON_DATE <dttm>, YEAR <dbl>, MONTH <dbl>, DAY_OF_WEEK <chr>,
#   HOUR <dbl>, UCR_PART <chr>, STREET <chr>, Lat <dbl>, Long <dbl>,
#   Location <chr>
```

Boston Crime Data Set

```
names(crime)
```

```
 [1] "INCIDENT_NUMBER"      "OFFENSE_CODE"         "OFFENSE_CODE_GROUP"
 [4] "OFFENSE_DESCRIPTION" "DISTRICT"             "REPORTING_AREA"
 [7] "SHOOTING"             "OCCURRED_ON_DATE"     "YEAR"
[10] "MONTH"                "DAY_OF_WEEK"          "HOUR"
[13] "UCR_PART"             "STREET"               "Lat"
[16] "Long"                 "Location"
```

display the structure of an object

Boston Crime Data Set

```
str(crime)
```

```
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    327820 obs.
 $ INCIDENT_NUMBER    : chr  "I182080058" "I182080053" "I182080052" "I18
 $ OFFENSE_CODE       : num  2403 3201 2647 413 3122 ...
 $ OFFENSE_CODE_GROUP : chr  "Disorderly Conduct" "Property Lost" "Other
 $ OFFENSE_DESCRIPTION: chr  "DISTURBING THE PEACE" "PROPERTY - LOST" "T
 $ DISTRICT           : chr  "E18" "D14" "B2" "A1" ...
 $ REPORTING_AREA     : num  495 795 329 92 36 351 NA 603 543 621 ...
 $ SHOOTING           : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ OCCURRED_ON_DATE   : POSIXct, format: "2018-10-03 20:13:00" "2018-08-
 $ YEAR               : num  2018 2018 2018 2018 2018 ...
 $ MONTH              : num  10 8 10 10 10 10 10 10 10 10 ...
 $ DAY_OF_WEEK        : chr  "Wednesday" "Thursday" "Wednesday" "Wednesd
 $ HOUR               : num  20 20 19 20 20 20 20 19 19 20 ...
 $ UCR_PART           : chr  "Part Two" "Part Three" "Part Two" "Part Or
 $ STREET             : chr  "ARLINGTON ST" "ALLSTON ST" "DEVON ST" "CAM
 $ Lat                : num  42.3 42.4 42.3 42.4 42.4 ...
 $ Long               : num  -71.1 -71.1 -71.1 -71.1 -71 ...
 $ Location           : chr  "(42.26260773, -71.12118637)" "(42.35211146
```

like a transposed version of print: columns run down the page, and data runs across.

Boston Crime Data Set

A little like str applied to a data frame but it tries to show you as much data as possible.

```
glimpse(crime)
```

```
Observations: 327,820
Variables: 17
$ INCIDENT_NUMBER     <chr> "I182080058", "I182080053", "I182080052", "I
$ OFFENSE_CODE        <dbl> 2403, 3201, 2647, 413, 3122, 1402, 3803, 330
$ OFFENSE_CODE_GROUP  <chr> "Disorderly Conduct", "Property Lost", "Othe
$ OFFENSE_DESCRIPTION <chr> "DISTURBING THE PEACE", "PROPERTY - LOST", "
$ DISTRICT            <chr> "E18", "D14", "B2", "A1", "A7", "C11", NA, "
$ REPORTING_AREA      <dbl> 495, 795, 329, 92, 36, 351, NA, 603, 543, 62
$ SHOOTING            <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA
$ OCCURRED_ON_DATE    <dttm> 2018-10-03 20:13:00, 2018-08-30 20:00:00, 2
$ YEAR                <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 20
$ MONTH               <dbl> 10, 8, 10, 10, 10, 10, 10, 10, 10, 10, 1
$ DAY_OF_WEEK         <chr> "Wednesday", "Thursday", "Wednesday", "Wedne
$ HOUR                <dbl> 20, 20, 19, 20, 20, 20, 20, 19, 19, 20, 19,
$ UCR_PART            <chr> "Part Two", "Part Three", "Part Two", "Part
$ STREET              <chr> "ARLINGTON ST", "ALLSTON ST", "DEVON ST", "C
$ Lat                 <dbl> 42.26261, 42.35211, 42.30813, 42.35945, 42.3
$ Long                <dbl> -71.12119, -71.13531, -71.07693, -71.05965,
```

Total crimes by day of week

```
crime %>% group_by(DAY_OF_WEEK) %>% summarise(count=n()) %>%
  mutate(prop=count/sum(count))
```

```
# A tibble: 7 x 3
  DAY_OF_WEEK count  prop
  <chr>       <int> <dbl>
1 Friday      49758 0.152
2 Monday      46970 0.143
3 Saturday    45969 0.140
4 Sunday      41374 0.126
5 Thursday    47872 0.146
6 Tuesday     47726 0.146
7 Wednesday   48151 0.147
```

alphabetic order

Total crimes by day of week descending by count

```
crime <- crime %>% mutate(Day_of_week =      new variable
                          fct_relevel(DAY_OF_WEEK,

        c("Monday","Tuesday","Wednesday",

        "Thursday","Friday","Saturday",
                                          "Sunday")))
crime %>% group_by(Day_of_week) %>% summarise(count=n()) %>%
  mutate(prop=count/sum(count))
```

```
# A tibble: 7 x 3
  Day_of_week count  prop
  <fct>       <int> <dbl>
1 Monday      46970 0.143
2 Tuesday     47726 0.146
3 Wednesday   48151 0.147
4 Thursday    47872 0.146
5 Friday      49758 0.152
6 Saturday    45969 0.140
7 Sunday      41374 0.126
```

Total crimes by month of year

```
crime %>% group_by(MONTH) %>% summarise(count=n()) %>%
  mutate(prop=count/sum(count)) %>% arrange(MONTH)
```

```
# A tibble: 12 x 3
   MONTH count   prop
   <dbl> <int>  <dbl>
 1     1 23625 0.0721
 2     2 21661 0.0661
 3     3 24156 0.0737
 4     4 24108 0.0735
 5     5 26242 0.0801
 6     6 30622 0.0934
 7     7 34640 0.106
 8     8 35137 0.107
 9     9 34023 0.104
10    10 26437 0.0806
11    11 23685 0.0723
12    12 23484 0.0716
```

Total crimes by month of year with abbreviations

abbreviation

```
month.abb
```

```
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov"
[12] "Dec"
```

```
crime <- crime %>% mutate(Month = month.abb[MONTH])
crime %>% select(MONTH, Month) %>% slice(1:5)
```

```
# A tibble: 5 x 2
  MONTH Month
  <dbl> <chr>
1    10 Oct
2     8 Aug
3    10 Oct
4    10 Oct
5    10 Oct
```

atomic vector

Total crimes by month of year

```
crime %>% group_by(Month) %>% summarise(count=n()) %>%
  mutate(prop=count/sum(count)) %>% arrange(Month)
```

```
# A tibble: 12 x 3
   Month count   prop
   <chr> <int>  <dbl>
 1 Apr   24108 0.0735
 2 Aug   35137 0.107
 3 Dec   23484 0.0716
 4 Feb   21661 0.0661
 5 Jan   23625 0.0721
 6 Jul   34640 0.106
 7 Jun   30622 0.0934
 8 Mar   24156 0.0737
 9 May   26242 0.0801
10 Nov   23685 0.0723
11 Oct   26437 0.0806
12 Sep   34023 0.104
```

Using factors and forcats

```
courses = c("MATH 203", "MATH 204", "MATH 208", "MATH 324",
            "MATH 423", "MATH 447",
            "MATH 523", "MATH 525", "MATH 533", "MATH 545")
courses
```

```
 [1] "MATH 203" "MATH 204" "MATH 208" "MATH 324" "MATH 423" "MATH 447"
 [7] "MATH 523" "MATH 525" "MATH 533" "MATH 545"
```

```
class(courses)
```

```
[1] "character"
```

as.numeric(courses)

=> error

Using factors and forcats

generic

```
courses_fct = factor(courses)
class(courses_fct)
```

atomic

```
[1] "factor"
```

```
mode(courses_fct)
```

```
[1] "numeric"
```

because

```
attributes(courses_fct)
```

```
$levels
 [1] "MATH 203" "MATH 204" "MATH 208" "MATH 324" "MATH 423" "MATH 447"
 [7] "MATH 523" "MATH 525" "MATH 533" "MATH 545"

$class
[1] "factor"
```

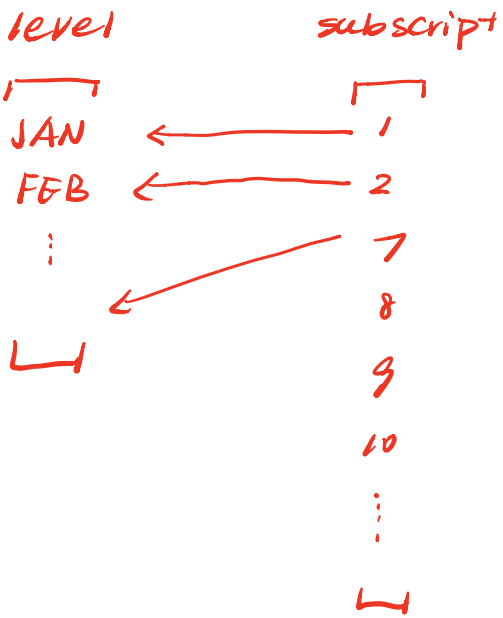
1
2
3
⋮

as.numeric(course_fct)

1 2 3 4 5 ...

reorder ?

Factors in R



Total crimes by day of week redux

```
crime <- crime %>% mutate(Day_of_week =
                            fct_relevel(DAY_OF_WEEK,

         c("Monday","Tuesday","Wednesday",

         "Thursday","Friday","Saturday",
                                          "Sunday")))
crime %>% group_by(Day_of_week) %>% summarise(count=n()) %>%
  mutate(prop=count/sum(count))
```

```
# A tibble: 7 x 3
  Day_of_week count  prop
  <fct>       <int> <dbl>
1 Monday      46970 0.143
2 Tuesday     47726 0.146
3 Wednesday   48151 0.147
4 Thursday    47872 0.146
5 Friday      49758 0.152
6 Saturday    45969 0.140
7 Sunday      41374 0.126
```

Total crimes by month of year as a factor

```
crime <- crime %>% mutate(Month = fct_relevel(Month,month.abb))
crime_by_month = crime %>% group_by(Month) %>%
  summarise(count=n()) %>%
  mutate(prop=count/sum(count)) %>% arrange(Month)
crime_by_month
```

```
# A tibble: 12 x 3
   Month count   prop
   <fct> <int>  <dbl>
 1 Jan   23625 0.0721
 2 Feb   21661 0.0661
 3 Mar   24156 0.0737
 4 Apr   24108 0.0735
 5 May   26242 0.0801
 6 Jun   30622 0.0934
 7 Jul   34640 0.106
 8 Aug   35137 0.107
 9 Sep   34023 0.104
10 Oct   26437 0.0806
11 Nov   23685 0.0723
12 Dec   23484 0.0716
```

Too much dessert using count tables (by count)

a month a row

```
ggplot(crime_by_month,aes(x="",y=count,
                          fill=Month)) +
  geom_bar(stat="identity") + coord_polar("y",start=0) +
  theme(axis.title.y =element_blank(),
        axis.text.y =element_blank(),
        axis.ticks.y=element_blank())
```

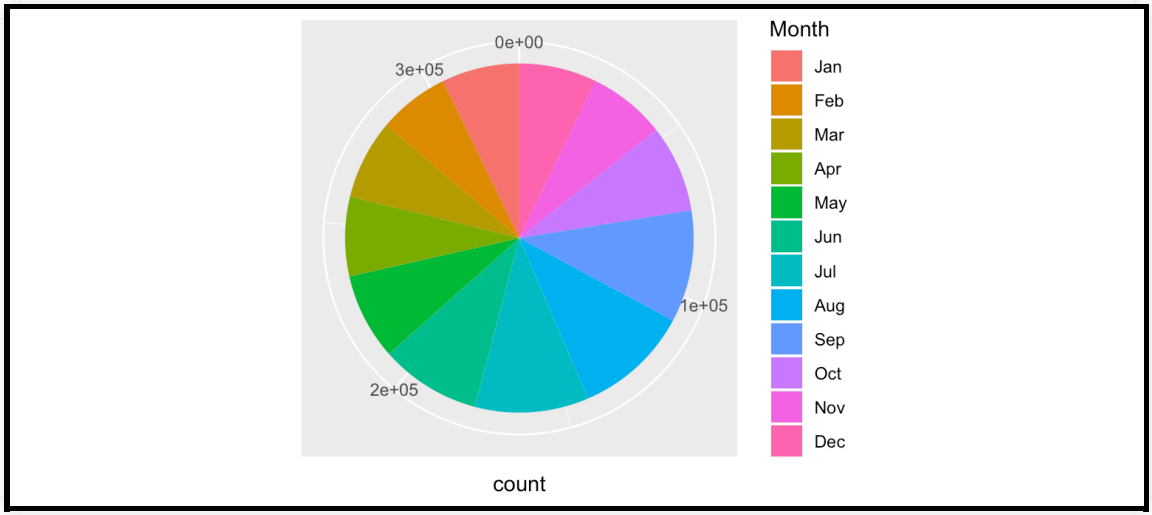
by default take fill variable count # rows they appear.

pie chart is a bar chart in a polar coord.

geom_bar categorical data

geom_histogram() for continuous data

stat_bin(), which bins data in ranges and counts the cases in each range. It differs from stat_count, which counts the number of cases at each x position.

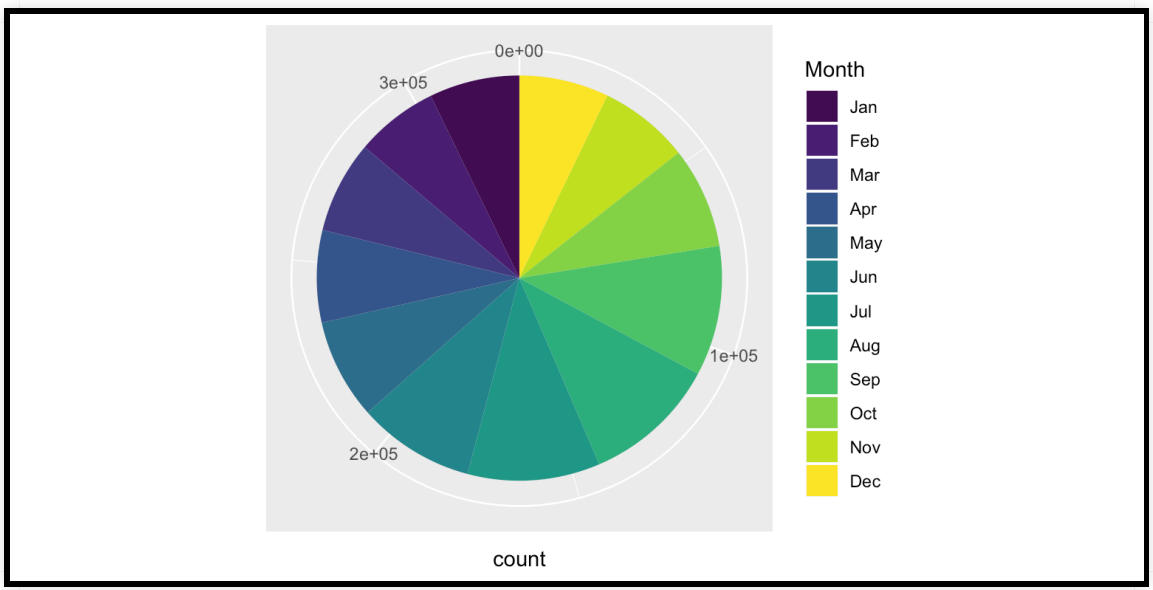

0e+00
3e+05
1e+05
2e+05
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
count

Too much dessert using count tables (by prop)

too much dessert using raw data

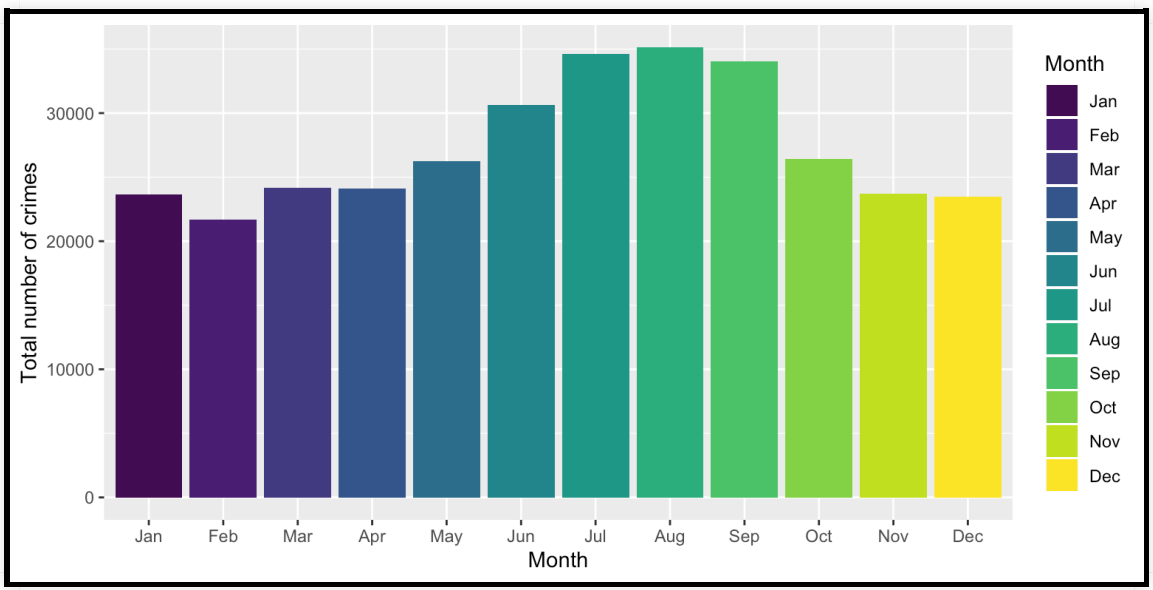
original dataset

```
ggplot(crime,aes(x=factor(1),fill=Month)) +
  geom_bar() + coord_polar("y",start=0) +
  scale_fill_viridis_d()   +
  theme(axis.title.y =element_blank(),
        axis.text.y =element_blank(),
        axis.ticks.y=element_blank())
```



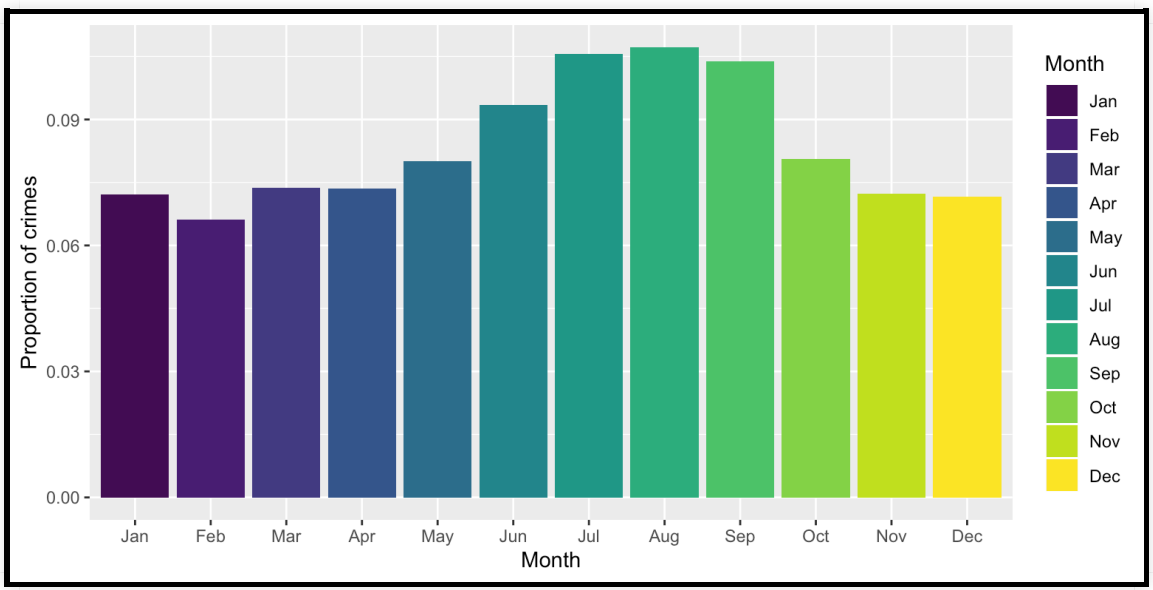
Maybe breakfast?

```
ggplot(crime,aes(x=Month,fill=Month)) +
  geom_bar() + scale_fill_viridis_d() + ylab("Total number of crimes")
```



Maybe breakfast?

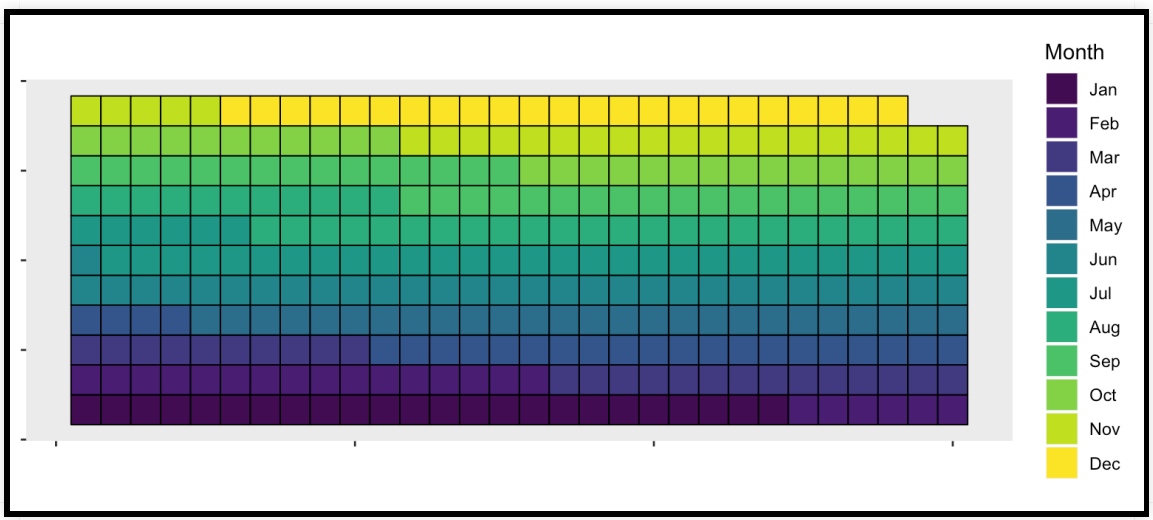
```
ggplot(crime,aes(x=Month,fill=Month)) +
  geom_bar(aes(y=..count../sum(..count..))) + scale_fill_viridis_d() +
  ylab("Proportion of crimes")
```



Do you like waffles?

Do you like waffles?

```
library(waffle)
library(hrbrthemes)
crime_by_month <- crime_by_month %>%
  mutate(waffle_count = round(count/1000))
ggplot(crime_by_month, aes(fill=Month,values=waffle_count)) +
  geom_waffle(n_rows=30,size = 0.33, flip=TRUE) +  coord_equal() +
  scale_fill_viridis_d() + theme_enhance_waffle()
```



Looking at crimes by type of crime (group)

```
crime %>% group_by(OFFENSE_CODE_GROUP) %>%
  summarise(count=n()) %>%
  mutate(prop=count/sum(count)) %>% arrange(desc(prop))
```

```
# A tibble: 67 x 3
   OFFENSE_CODE_GROUP              count   prop
   <chr>                           <int>  <dbl>
 1 Motor Vehicle Accident Response 38134 0.116
 2 Larceny                         26670 0.0814
 3 Medical Assistance              24226 0.0739
 4 Investigate Person              19176 0.0585
 5 Other                           18612 0.0568
 6 Drug Violation                  17037 0.0520
 7 Simple Assault                  16263 0.0496
 8 Vandalism                       15810 0.0482
 9 Verbal Disputes                 13478 0.0411
10 Towed                           11632 0.0355
# … with 57 more rows
```

Looking at crimes by type of crime (group)

```
off_code_counts <- crime %>% group_by(OFFENSE_CODE_GROUP) %>%
  summarise(count=n())  %>% mutate(prop=count/sum(count))
ggplot(off_code_counts,aes(x=OFFENSE_CODE_GROUP,fill=OFFENSE_CODE_GROUP)
         +
  geom_bar(stat="identity",aes(y=prop))+ scale_fill_viridis_d() +
  ylab("Proportion of crimes")+
  theme(axis.title.x =element_blank(),
        axis.text.x =element_blank(),
        axis.ticks.x=element_blank())
```


ravated Assault
raft
on
embly or Gathering Violations
o Theft
o Theft Recovery
istics
ogical Threat
nb Hoax
glary - No Property Taken
nmercial Burglary
nfidence Games
nterfeiting
minal Harassment
orderly Conduct
g Violation
Evading Fare
Explosives
Fire Related Reports
Firearm Discovery
Firearm Violations
Fraud
Gambling
Harassment
Harbor Related Incidents
HOME INVASION
Homicide
HUMAN TRAFFICKING
HUMAN TRAFFICKING - INVOLUNTARY SERVITUDE
Investigate Person
INVESTIGATE PERSON
Investigate Property
Larceny
Larceny From Motor Vehicle
License Plate Related Incidents
License Violation
Liquor Violation
Manslaughter
Medical Assistance
Missing Person Located
Missing Person Reported
Motor Vehicle Accident Response
Offenses Against Child / Family
Operating Under the Influence
Other
Other Burglary
Phone Call Complaints
Police Service Incidents
Proper
Proper
Proper
Prostitu
Recove
Reside
Restra
Robber
Search
Service
Simple
Towed
Vandal
Verbal
Violatio
Warran

Looking at crimes by type of crime (group)

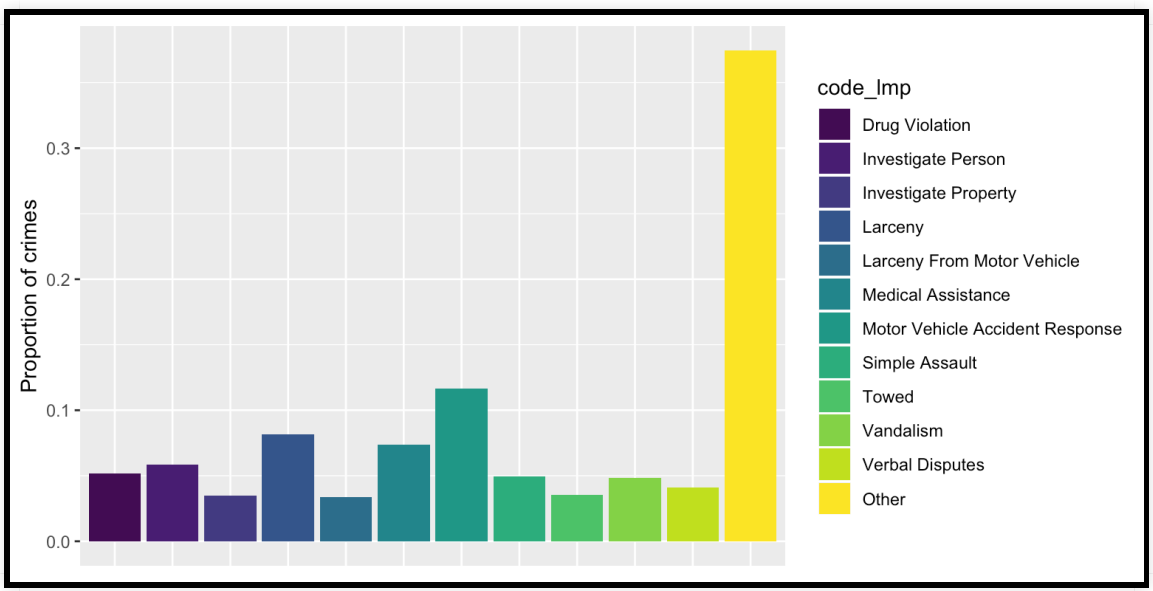
lump together least/most common factor level into "other"

```
crime <- crime %>%
  mutate(code_lmp = fct_lump(OFFENSE_CODE_GROUP,12))
off_code_counts_lmp <- crime %>%
  group_by(code_lmp) %>%  count() %>%  ungroup() %>%
  mutate(prop = n/sum(n)) %>% arrange(n)
off_code_counts_lmp
```

```
# A tibble: 12 x 3
   code_lmp                          n   prop
   <fct>                         <int>  <dbl>
 1 Larceny From Motor Vehicle    11120 0.0339
 2 Investigate Property          11443 0.0349
 3 Towed                         11632 0.0355
 4 Verbal Disputes               13478 0.0411
 5 Vandalism                     15810 0.0482
 6 Simple Assault                16263 0.0496
 7 Drug Violation                17037 0.0520
 8 Investigate Person            19176 0.0585
 9 Medical Assistance            24226 0.0739
10 Larceny                       26670 0.0814
11 Motor Vehicle Accident Response  38134 0.116
12 Other                        122831 0.375
```

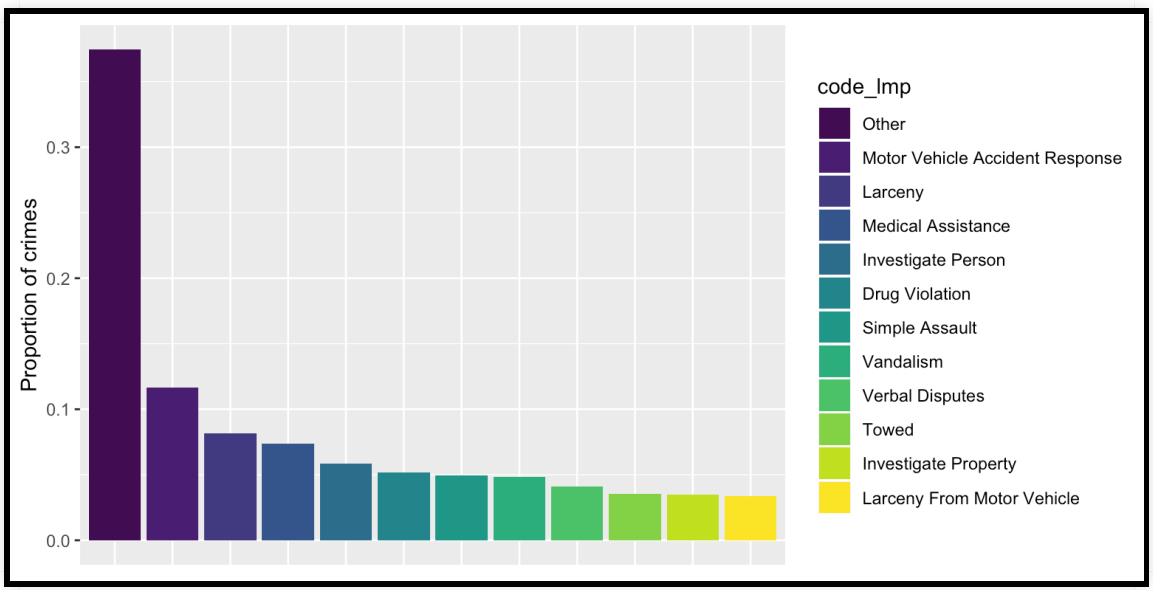
Looking at crimes by type of crime (group)

```
ggplot(off_code_counts_lmp,aes(x=code_lmp,fill=code_lmp)) +
  geom_bar(stat="identity",aes(y=prop))+
  scale_fill_viridis_d() +ylab("Proportion of crimes")  +
  theme(axis.title.x =element_blank(),axis.text.x =element_blank(),
        axis.ticks.x=element_blank())
```



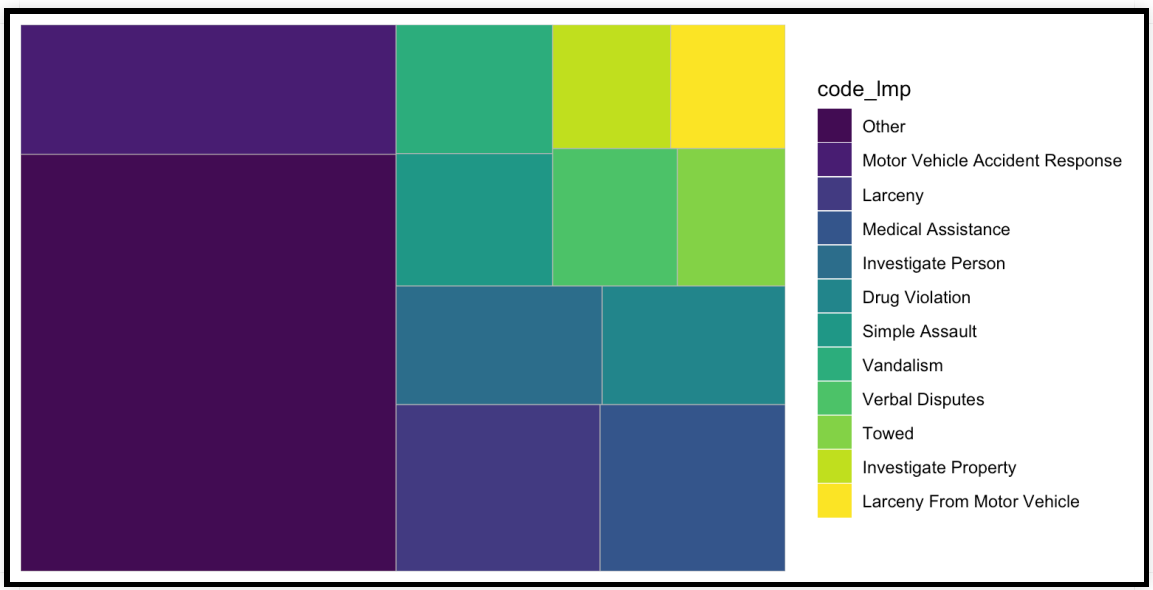
Looking at crimes by type of crime (group)

```
off_code_counts_lmp <- off_code_counts_lmp %>%
  mutate(code_lmp = fct_reorder(code_lmp, n,.desc=TRUE))
ggplot(off_code_counts_lmp,aes(x=code_lmp,fill=code_lmp)) +
  geom_bar(stat="identity",aes(y=prop))+ scale_fill_viridis_d() +
  ylab("Proportion of crimes")  +  theme(axis.title.x =element_blank(),
        axis.text.x =element_blank(),axis.ticks.x=element_blank())
```



Would you like them in a tree?

```
# install.packages("treemapify")
library(treemapify)
ggplot(off_code_counts_lmp, aes(area = n, fill = code_lmp)) +
  geom_treemap() + scale_fill_viridis_d()
```



Sidenote: Saving plots as objects

```
off_code_counts <- off_code_counts %>%
  mutate(OFFENSE_CODE_GROUP = fct_reorder(OFFENSE_CODE_GROUP,
                                          count,.desc=TRUE))
p <- ggplot(off_code_counts,
       aes(area = count, fill = OFFENSE_CODE_GROUP)) +
  geom_treemap()
class(p)
```

```
[1] "gg"     "ggplot"
```

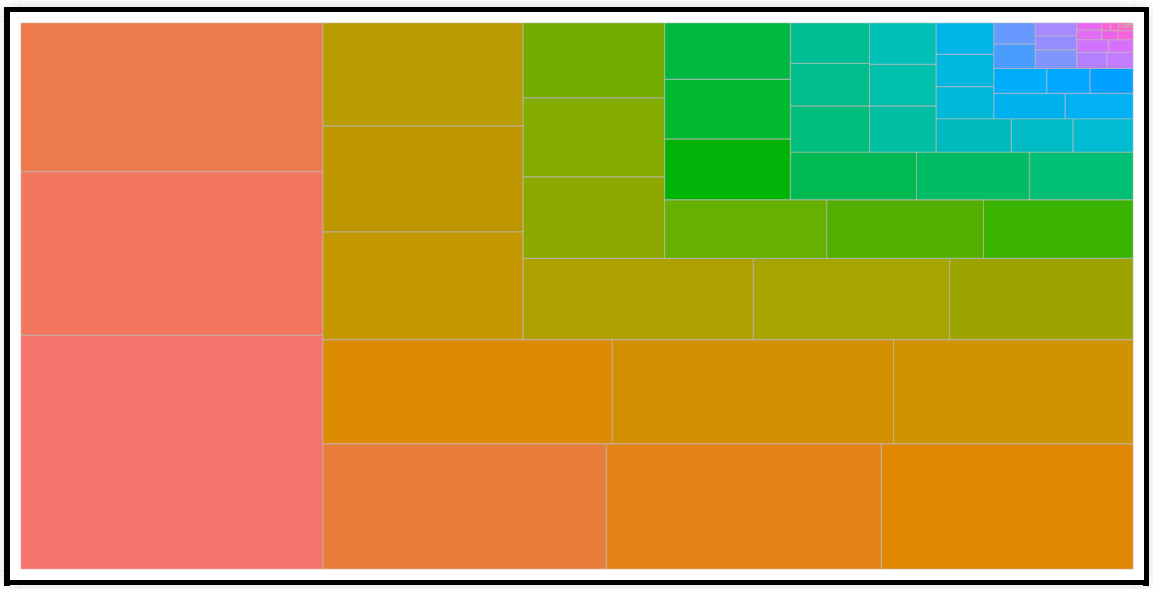
```
attributes(p)
```

```
$names
[1] "data"        "layers"      "scales"      "mapping"     "theme"
[6] "coordinates" "facet"       "plot_env"    "labels"

$class
[1] "gg"     "ggplot"
```

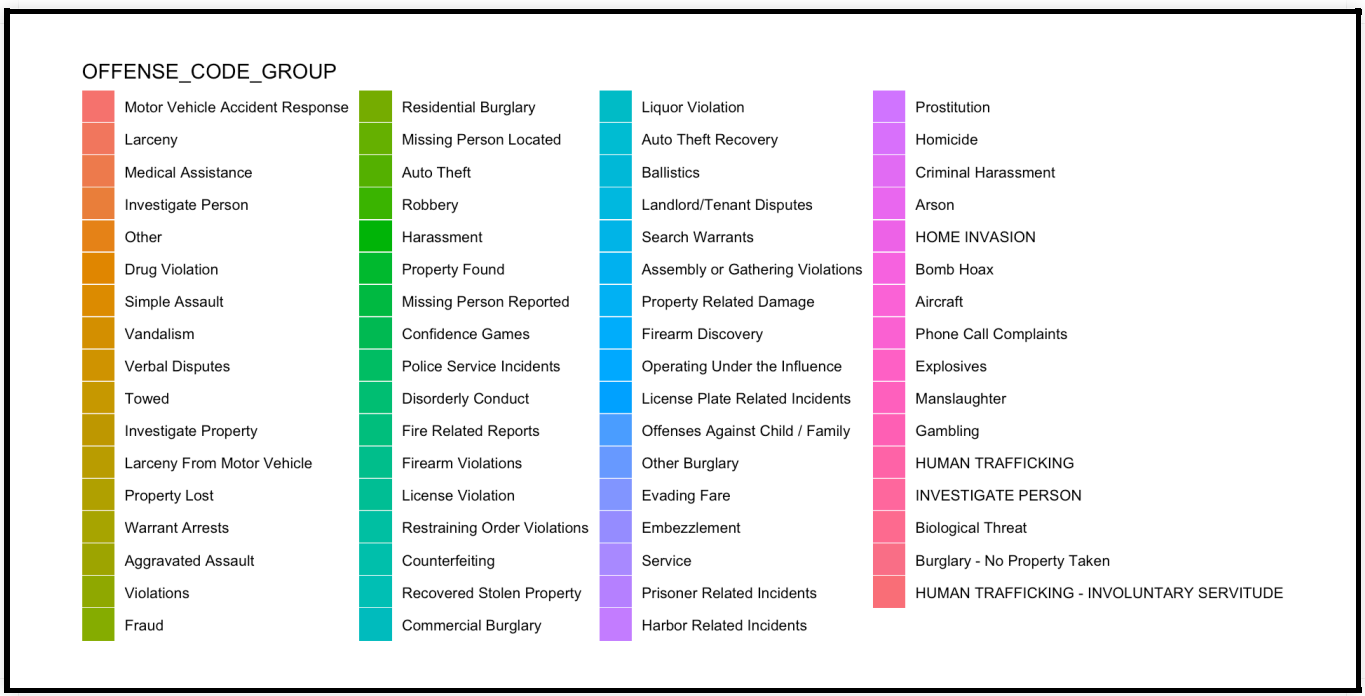
Would you like them all in a tree?

```
p + theme(legend.position="none")
```



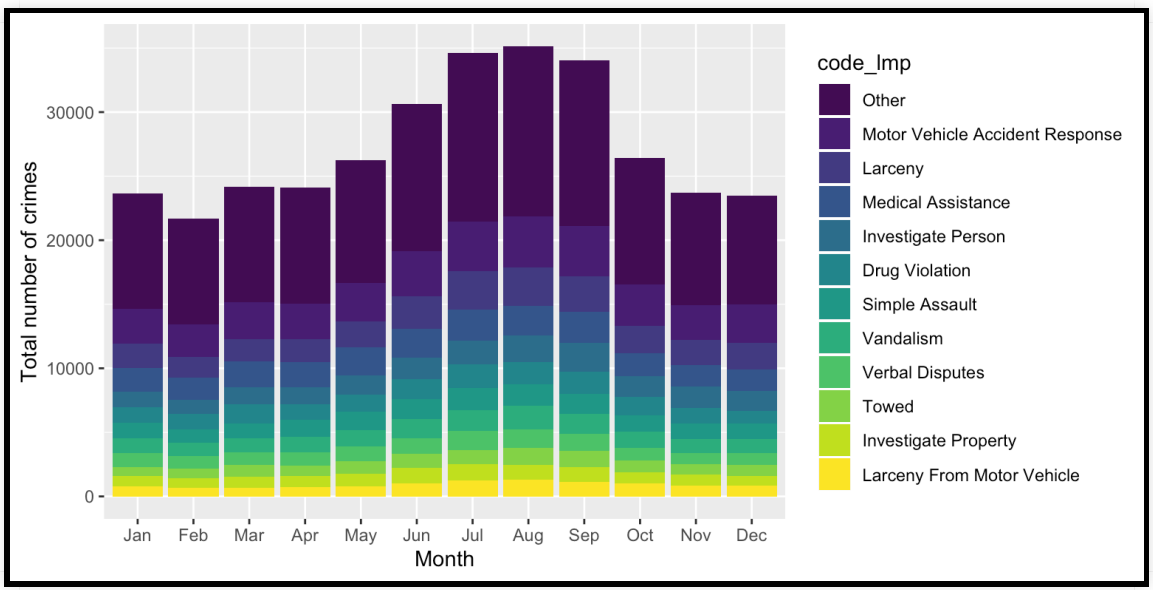
Would you like them all in a tree?

```
#install.packages("ggpubr")
library(ggpubr)
as_ggplot(get_legend(p+theme(legend.text=element_text(size=8))))
```



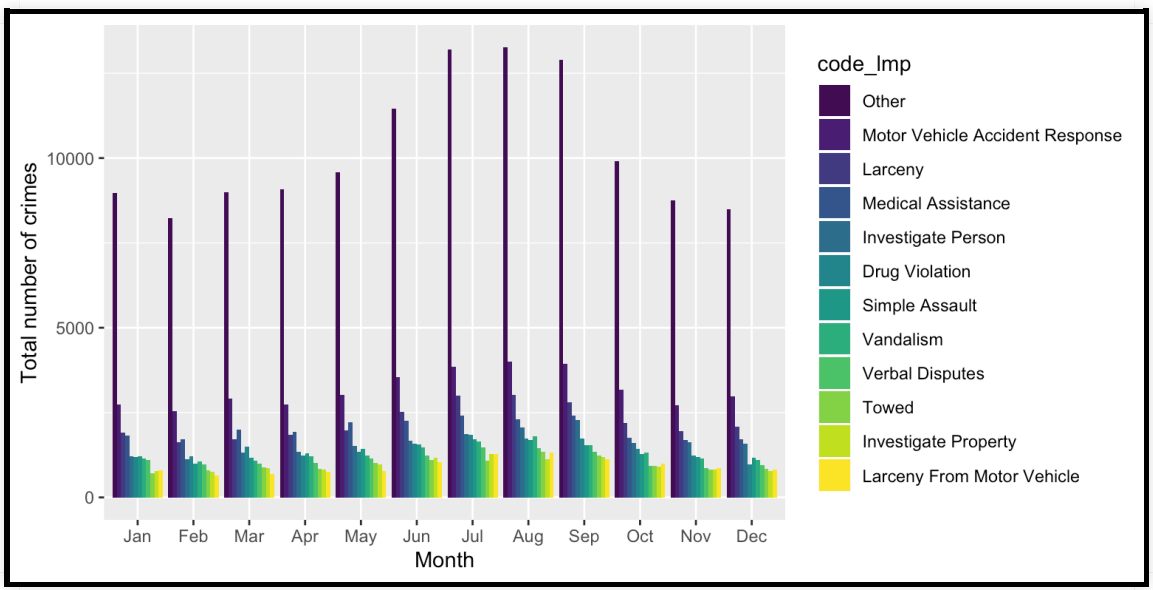
More than one qualitative factor: Stacking bars

```
crime = crime %>% mutate(code_lmp = fct_infreq(code_lmp))
ggplot(crime,aes(x=Month,fill=code_lmp)) +
  geom_bar() + scale_fill_viridis_d() + ylab("Total number of crimes")
```



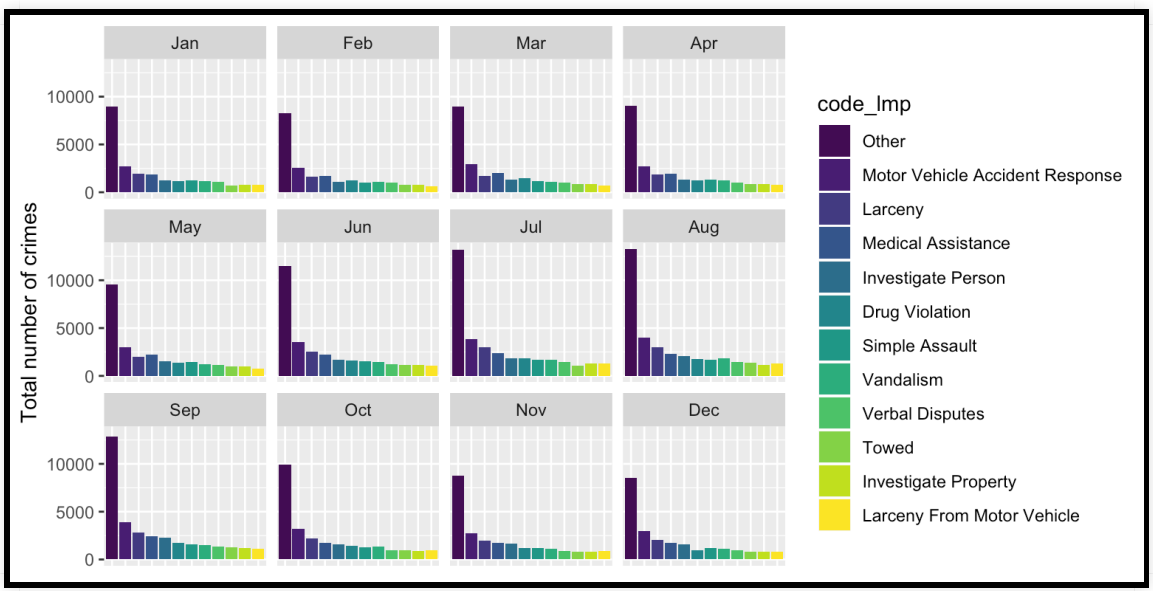
Dodging bars

```
ggplot(crime,aes(x=Month,fill=code_lmp)) +
  geom_bar(position="dodge") + scale_fill_viridis_d() +
  ylab("Total number of crimes")
```



Faceting bars

```
ggplot(crime,aes(x=code_lmp,fill=code_lmp)) +
  geom_bar(position="dodge") + facet_wrap(~Month) +
  scale_fill_viridis_d() + ylab("Total number of crimes")+
  theme(axis.title.x =element_blank(),
        axis.text.x =element_blank(),axis.ticks.x=element_blank())
```

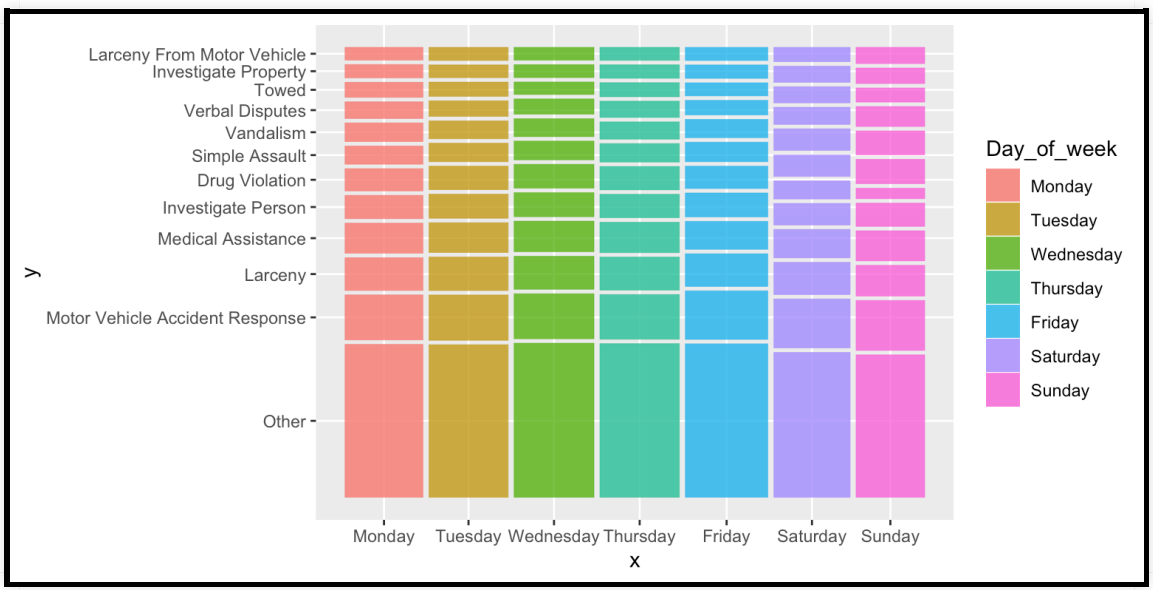


Mosaic plots

If the distribution of crimes differs over the days.

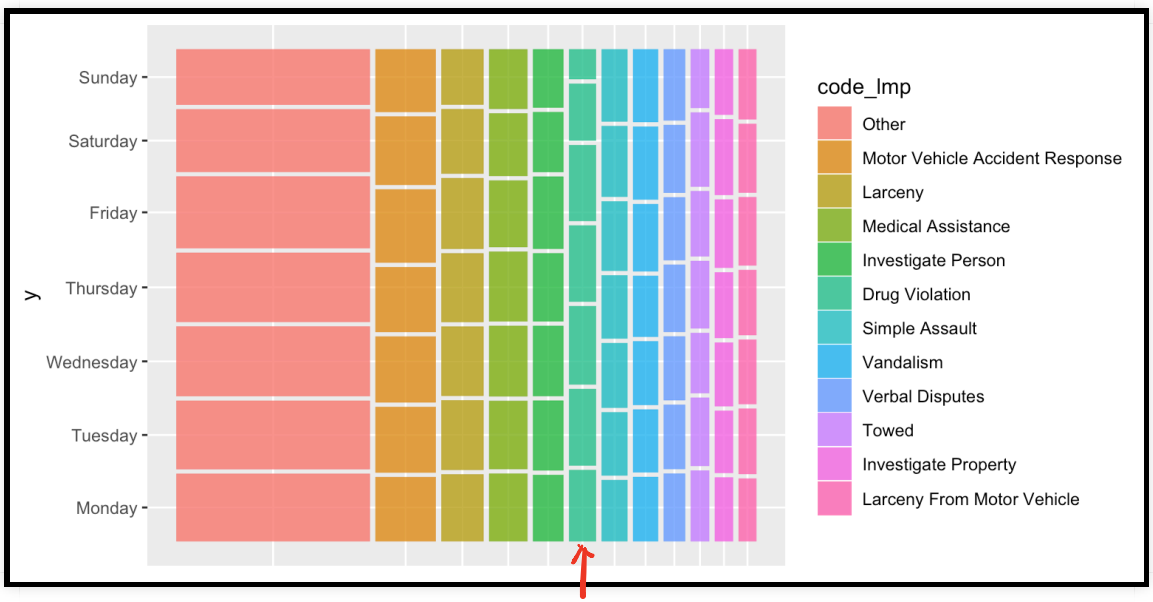
12 x 7 = 84 levels

```
library(ggmosaic)
ggplot(crime) + geom_mosaic(aes(x=product(code_lmp,Day_of_week),
                                fill=Day_of_week))
```



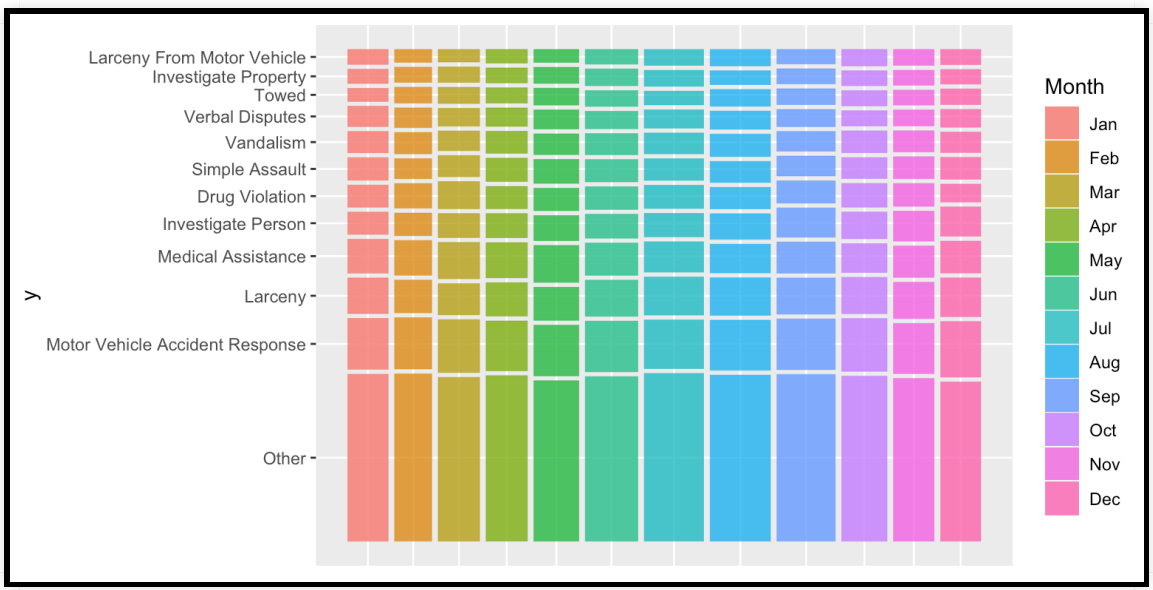
Mosaic plots

```
ggplot(crime) + geom_mosaic(aes(x=product(Day_of_week,code_lmp),
                                fill=code_lmp))+
  theme(axis.title.x =element_blank(),axis.text.x =element_blank(),
        axis.ticks.x=element_blank())
```



Mosaic plots

```
ggplot(crime) +
  geom_mosaic(aes(x=product(code_lmp,Month),fill=Month))+
  theme(axis.title.x =element_blank(),axis.text.x =element_blank(),
        axis.ticks.x=element_blank())
```



Date/time formats

POSIXlt

```
crime %>% select(OCCURRED_ON_DATE) %>% head(20)
```

```
# A tibble: 20 x 1
   OCCURRED_ON_DATE
   <dttm>
 1 2018-10-03 20:13:00
 2 2018-08-30 20:00:00
 3 2018-10-03 19:20:00
 4 2018-10-03 20:00:00
 5 2018-10-03 20:49:00
 6 2018-10-02 20:40:00
 7 2018-10-03 20:16:00
 8 2018-10-03 19:32:00
 9 2018-10-03 19:27:00
10 2018-10-03 20:00:00
11 2018-10-03 19:33:00
12 2018-10-01 20:00:00
13 2018-10-03 17:18:00
14 2018-10-03 08:00:00
15 2018-10-03 19:58:00
```

POSIXlt

```
crime %>% summarise(min=min(OCCURRED_ON_DATE),
                    med = median(OCCURRED_ON_DATE),
                    max = max(OCCURRED_ON_DATE))
```

```
# A tibble: 1 x 3
  min                 med                 max
  <dttm>              <dttm>              <dttm>
1 2015-06-15 00:00:00 2017-02-14 15:49:00 2018-10-03 20:49:00
```

generic vector
anonymous

Everything coming from the pipe

```
crime %>% pull(OCCURRED_ON_DATE) %>%  class(.)
```

return a vector for tbl_df

```
[1] "POSIXct" "POSIXt"
```

```
crime %>% pull(OCCURRED_ON_DATE) %>% head(10) %>%  as.numeric(.)
```

```
 [1] 1538597580 1535659200 1538594400 1538596800 1538599740 1538512800
 [7] 1538597760 1538595120 1538594820 1538596800
```

number of second since 1970.01.01

Make your own

tidyverse

```
library(lubridate)
my_date <- "1991-05-25"
class(my_date)
```

different format?

```
[1] "character"
```

character string in year month day format

```
my_date <- ymd(my_date)
my_date
```

```
[1] "1991-05-25"
```

```
class(my_date)
```

```
[1] "Date"
```

Make your own

timezone

```
other_date = ymd_hms("2009-05-02 02:57:00", tz="America/Montreal")
other_date
```

```
[1] "2009-05-02 02:57:00 EDT"
```

lubridate arithmetic

```
my_date + days( (-2):2)
```

```
[1] "1991-05-23" "1991-05-24" "1991-05-25" "1991-05-26" "1991-05-27"
```

```
my_date + months( (-2):2)
```

```
[1] "1991-03-25" "1991-04-25" "1991-05-25" "1991-06-25" "1991-07-25"
```

```
now()
```

```
[1] "2019-10-04 10:22:07 EDT"
```

lubridate arithmetic

```
now() - other_date
```

```
Time difference of 3807.309 days
```

365 ? OR true year ?

```
interval(other_date,now())/years(1)
```

```
[1] 10.42434
```

Counts of crimes by date

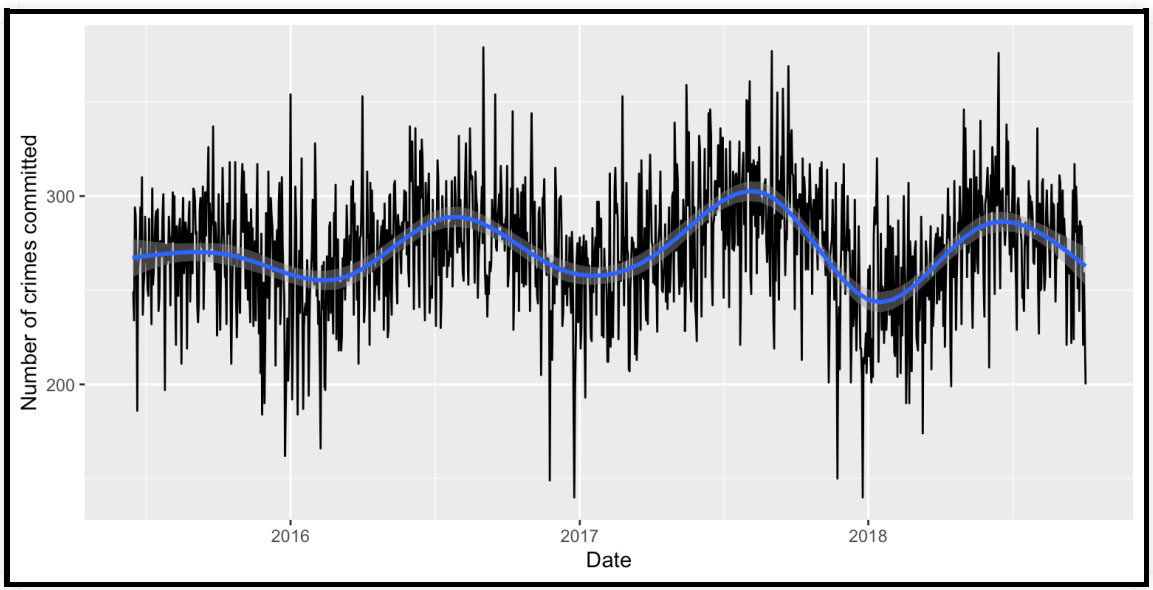
```
by_date_tbl = crime %>% mutate(date_only = date(OCCURRED_ON_DATE)) %>%
  group_by(date_only) %>%
  summarise(count=n())
by_date_tbl %>% arrange(desc(count)) %>% head(5)
```

extract only the date

```
# A tibble: 5 x 2
  date_only  count
  <date>     <int>
1 2016-09-01   379
2 2017-09-01   377
3 2018-06-15   376
4 2017-09-22   369
5 2017-08-04   361
```

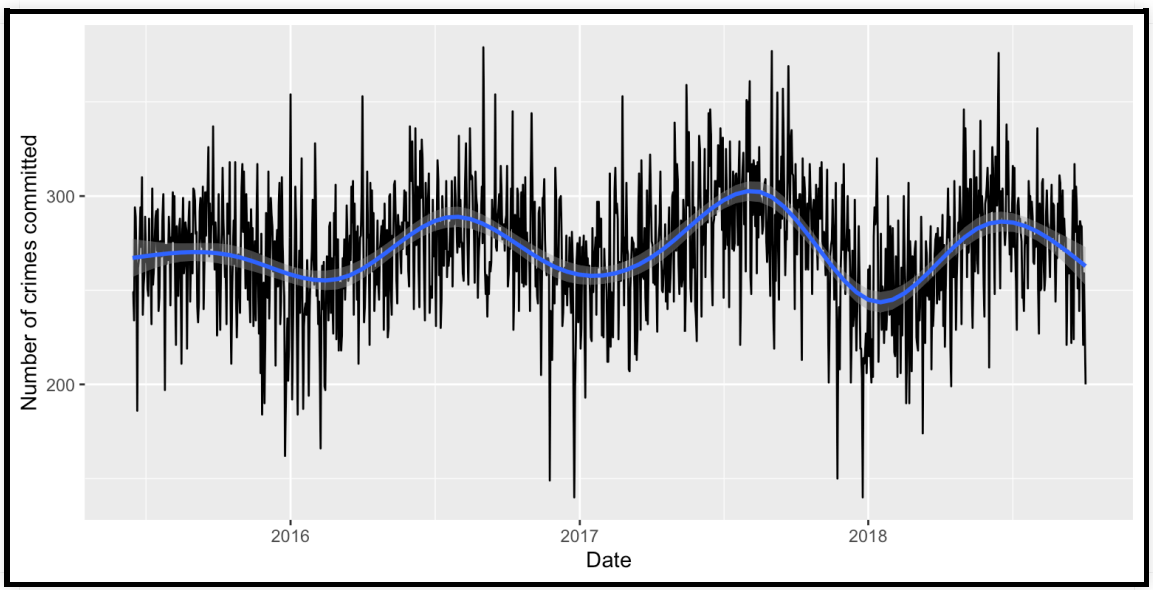
Counts of crimes by date

```
ggplot(by_date_tbl,
       aes(x=date_only, y= count)) + geom_line() +
  labs(x="Date",y="Number of crimes committed") + geom_smooth()
```



Counts of crimes by date

```
ggplot(by_date_tbl,
       aes(x=date_only, y= count)) + geom_line() +
  labs(x="Date",y="Number of crimes committed") + geom_smooth()
```



Counts of crimes by month: Method 1

```
by_month_tbl  = crime %>% group_by(YEAR,Month) %>%
  summarise(count=n())
by_month_tbl %>% arrange(desc(count)) %>% head(5)
```

```
# A tibble: 5 x 3
# Groups:   YEAR [2]
   YEAR Month count
  <dbl> <fct> <int>
1  2017 Aug    9209
2  2017 Jul    9077
3  2017 Jun    8990
4  2017 Sep    8950
5  2016 Aug    8940
```

Counts of crimes by month: Method 1

```
by_month_tbl = by_month_tbl %>% ungroup() %>%
  mutate(Month_Year=factor(interaction(Month,YEAR),
                           levels=interaction(Month,YEAR)))
Jan_levels = by_month_tbl %>%
                      filter(Month=="Jan") %>% pull(Month_Year)%>%
  unique(.)
tibble(Jan_levels,as.numeric(Jan_levels))
```

combine 2 columns with a dot in between.

order

```
# A tibble: 3 x 2
  Jan_levels `as.numeric(Jan_levels)`
  <fct>                         <dbl>
1 Jan.2016                          8
2 Jan.2017                         20
3 Jan.2018                         32
```

Counts of crimes by month: Method 1

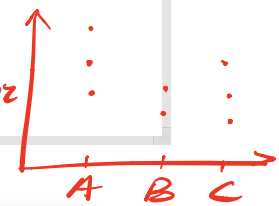
```
ggplot(by_month_tbl,aes(x=Month_Year,y=count,group=1)) +
  geom_point() +   geom_line(stat="summary",fun.y=sum) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x="Date",y="Count",title="Number of crimes by month") +
  geom_vline(xintercept=as.numeric(Jan_levels),
      col="red",linetype="dashed")
```

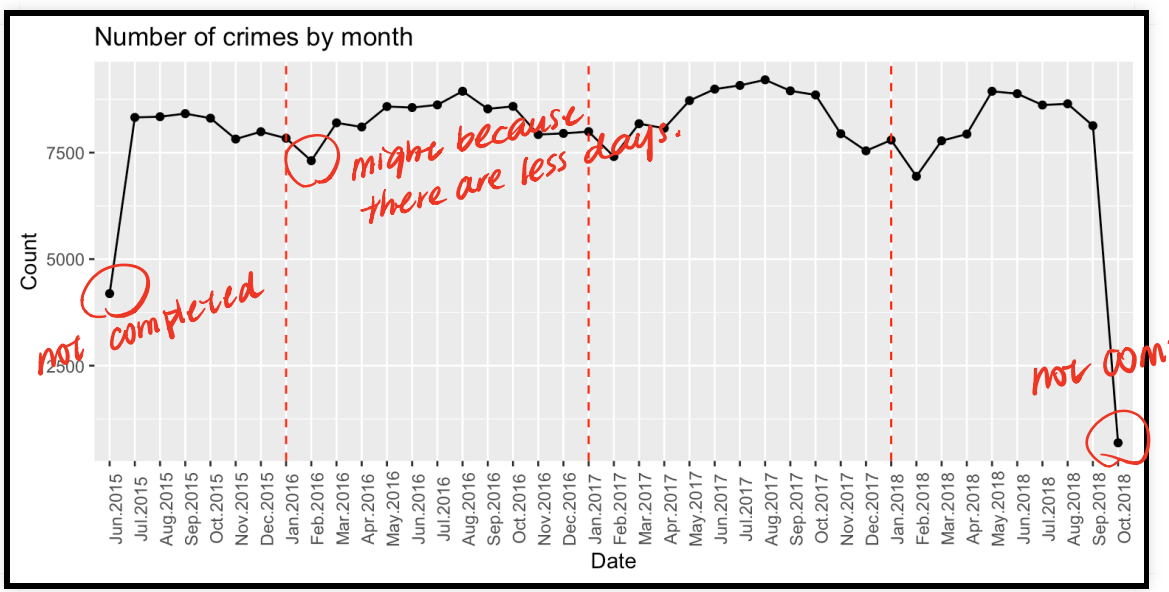
everything in the same group

how to connect points

doesn't matter

only 1 point





Counts of crimes by month: Method 2

```
crime = crime %>% mutate(First_of_month =
                         floor_date(OCCURRED_ON_DATE,unit="month"))
crime %>% slice(sample(x=1:n(),size=10)) %>%
  select(OCCURRED_ON_DATE,First_of_month)
```

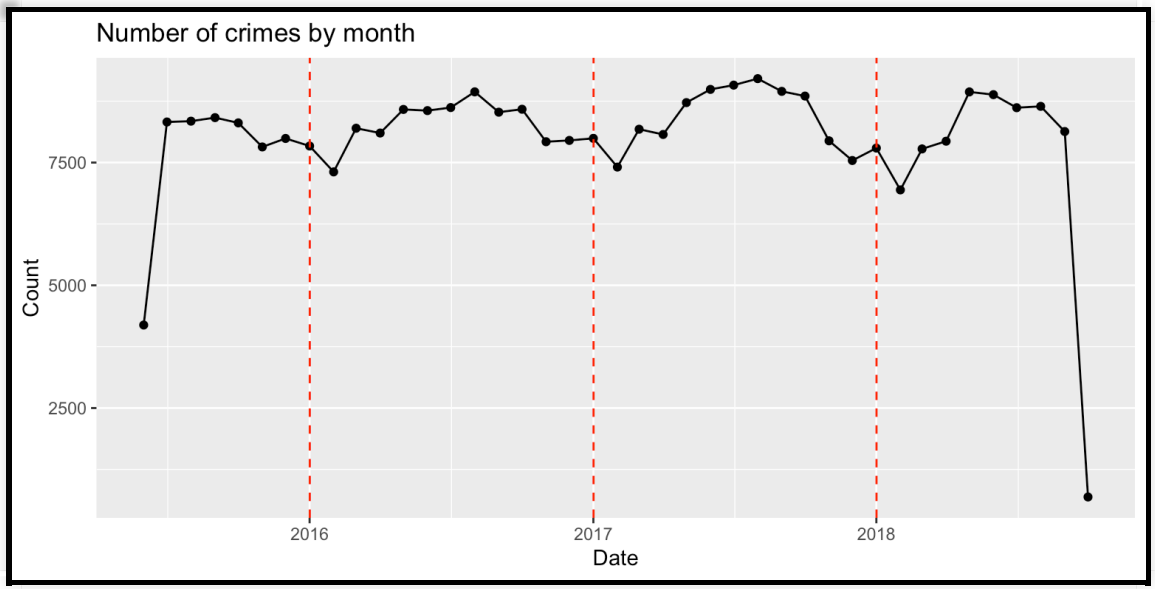
round down

difference between slice and filter

```
# A tibble: 10 x 2
   OCCURRED_ON_DATE    First_of_month
   <dttm>              <dttm>
 1 2018-09-12 13:49:00 2018-09-01 00:00:00
 2 2017-04-13 23:30:00 2017-04-01 00:00:00
 3 2015-11-19 05:12:00 2015-11-01 00:00:00
 4 2017-05-23 21:25:00 2017-05-01 00:00:00
 5 2016-05-08 11:59:00 2016-05-01 00:00:00
 6 2018-05-13 12:31:00 2018-05-01 00:00:00
 7 2016-04-18 18:30:00 2016-04-01 00:00:00
 8 2017-09-11 13:10:00 2017-09-01 00:00:00
 9 2016-05-22 15:15:00 2016-05-01 00:00:00
10 2018-02-23 15:00:00 2018-02-01 00:00:00
```

Counts of crimes by month: Method 2

```
by_month_tbl2  = crime %>% group_by(First_of_month) %>%
  summarise(count=n())
ggplot(by_month_tbl2,aes(x=First_of_month,y=count)) +
  geom_point() +   geom_line() +
  labs(x="Date",y="Count",title="Number of crimes by month") +
  geom_vline(xintercept=
               as.POSIXct(c("2016-01-01","2017-01-01","2018-01-01")),
             col="red",linetype="dashed")
```




```
# Missing values, NaN, and Inf
```

Missing values, Nan, Inf

different

```
1/0
```

```
[1] Inf
```

```
exp(-Inf)
```

```
[1] 0
```

```
0/0
```

```
[1] NaN
```

```
sqrt(-1)
```

```
[1] NaN
```

```
sqrt(as.complex(-1))
```

- c(1, 2, 3, -Inf) + c(NA, Inf, NaN, Inf)
  = NA Inf NaN NaN
- as.numeric("My missing value")
  NA
- as.numeric(factor("My"))
  1
- c(1, "3")
  "1", "3"
- as.numeric(c(1, "3"))
  1, 3
- as.numeric(c(1, "3", "cat"))
  1, 3, NA

```
[1] 0+1i
```

Missing in your dataset

composite function

0 false 1 true

#missing value.

```
crime %>% summarise_all(list(~sum(is.na(.)))) %>%
  pivot_longer(cols=everything(),names_to = "Variable")
```

```
# A tibble: 21 x 2
   Variable             value
   <chr>                <int>
 1 INCIDENT_NUMBER          0
 2 OFFENSE_CODE             0
 3 OFFENSE_CODE_GROUP       0
 4 OFFENSE_DESCRIPTION      0
 5 DISTRICT              1774
 6 REPORTING_AREA       20920
 7 SHOOTING                 0
 8 OCCURRED_ON_DATE         0
 9 YEAR                     0
10 MONTH                    0
# … with 11 more rows
```

Missing in your dataset

```
crime %>% summarise_all(list(~sum(is.na(.)))) %>%
  pivot_longer(cols=everything(),names_to = "Variable") %>%
  filter(value>0)
```

```
# A tibble: 6 x 2
  Variable       value
  <chr>          <int>
1 DISTRICT        1774
2 REPORTING_AREA 20920
3 UCR_PART          93
4 STREET         10977
5 Lat            20632
6 Long           20632
```

Removing missing from your dataset

```
crime_no_na <- crime %>% drop_na()
crime %>% summarise(n())
```

```
# A tibble: 1 x 1
   `n()`
   <int>
1 327820
```

```
crime_no_na %>% summarise(n())
```

```
# A tibble: 1 x 1
   `n()`
   <int>
1 304167
```

```
crime %>% drop_na(UCR_PART) %>% summarise(n())
```

specific

```
# A tibble: 1 x 1
   `n()`
   <int>
1 327727
```