# M&E plots in ggplot2

*Nathan Green, Imperial College London*

*07/09/2019*

### Distribution of AFP Cases by Month, Pakistan 2015-2019

Read in the required packages.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## 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
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.5.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

Read in data from Excel file `List of AFP Cases 2015-2019.xlsx`.

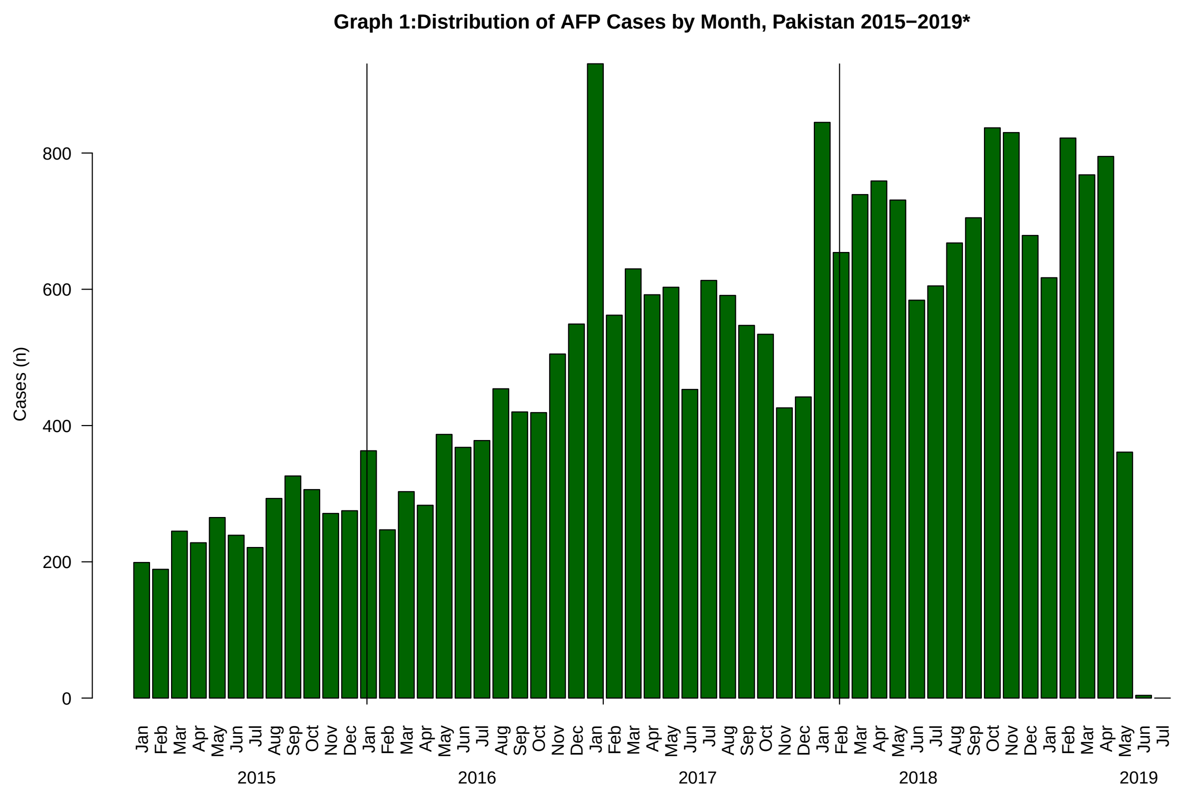
```
library(dataPakistan)

file_name <- system.file(package = "dataPakistan", "extdata", "List of AFP Cases 2015-2019.xlsx")
dat <- readxl::read_xlsx(file_name, sheet = "Data")
```

```
##############
# preprocess #
##############

# extract only the month from the date in `DENTER`
dat$month <- lubridate::month(dat$DENTER, label = TRUE)

# create a new column called `afp_cases` that is `TRUE` is `ALL DIAGNOSED` is recorded in `AFP`
dat$afp_cases <- dat$AFP == "ALL DIAGNOSED"
```

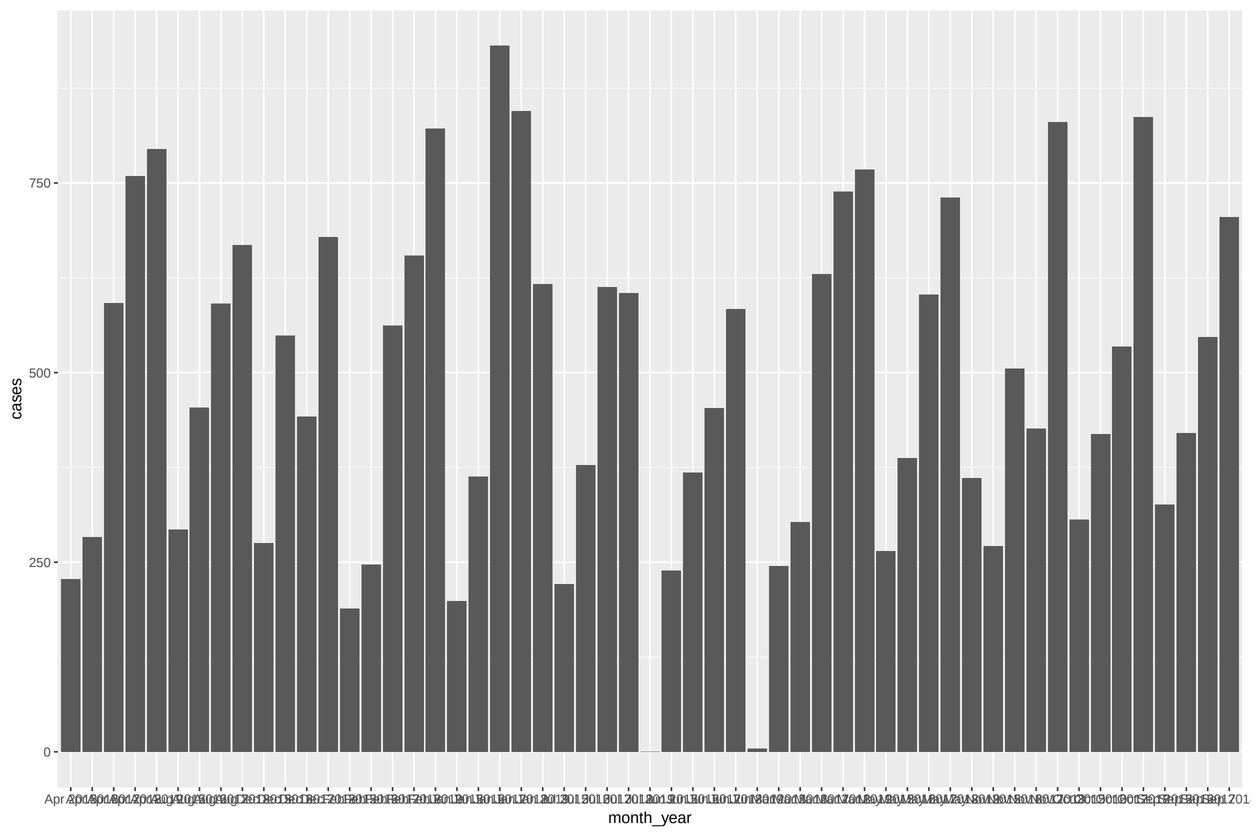
We're going to use the `dplyr` package functions to take aggregate statistics for months and years on onset.

```r
x <-
  dat %>%
  group_by(month, YRONSET) %>%                          # separate the data set in to subsets of each 
  summarise(cases = sum(afp_cases, na.rm = TRUE)) %>%   # within each of these subgroups calculate the
  mutate(month_year = paste(month, YRONSET)) %>%        # create a new column coombining month and yea
  arrange(YRONSET, month)                               # sort the rows by increasing year of onset an
```

```r
#########
# plots #
#########


# basic base R bar plot

barplot(height = x$cases,
        col = "darkgreen",    # colour the bars
        names.arg = x$month, # labels on the x-axis
        las = 2)             # resize the axis labels

# Annotate the x-axis with the year 2015 to 2019
mtext("2015", side = 1, line = 3, at = 8)
mtext("2016", side = 1, line = 3, at = 22)
mtext("2017", side = 1, line = 3, at = 36)
mtext("2018", side = 1, line = 3, at = 50)
mtext("2019", side = 1, line = 3, at = 64)

title(main = "Graph 1:Distribution of AFP Cases by Month, Pakistan 2015-2019*",
      ylab = "Cases (n)")

# add vertical lines at given times
abline(v = 15)
abline(v = 30)
abline(v = 45)
```



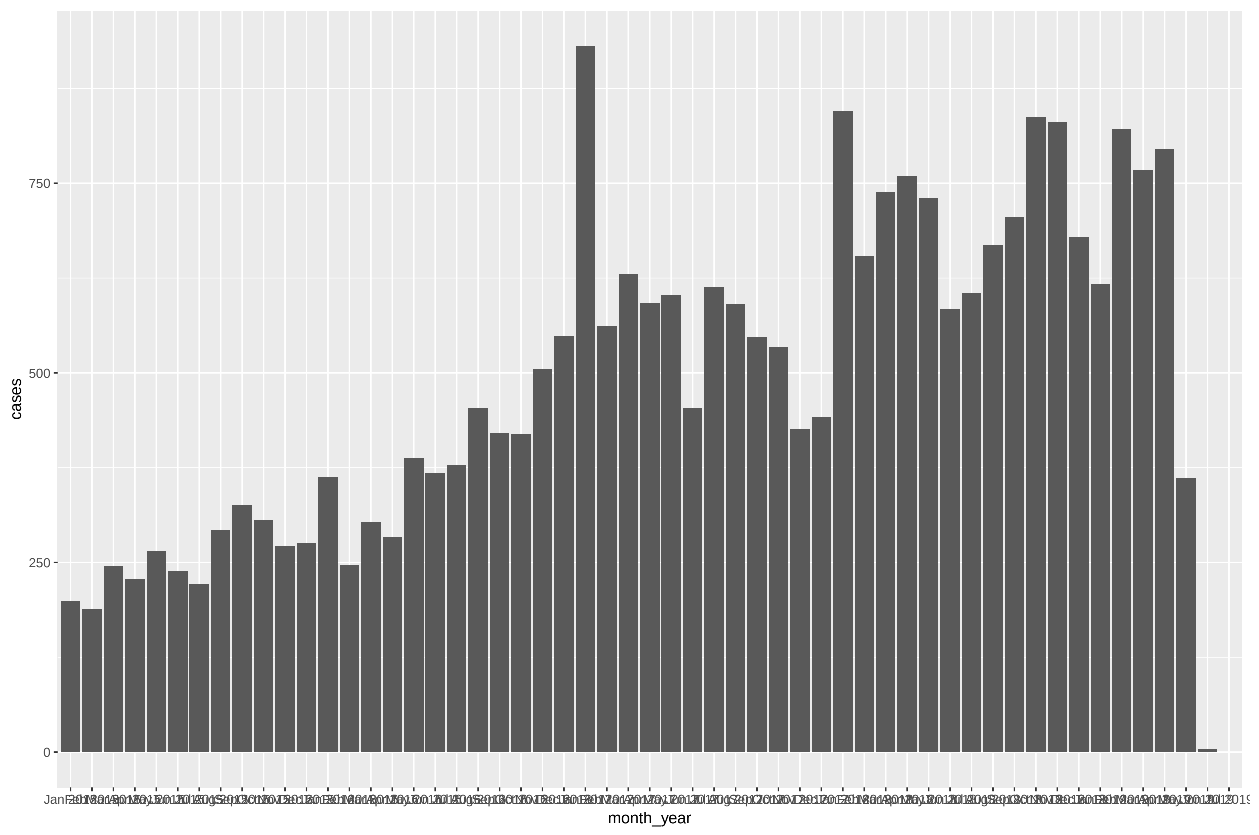
```
# ggplot version

# basic ggplot bar plot
plot0 <-
  ggplot(x, aes(x = month_year, y = cases)) +
  geom_bar(stat = "identity")

plot0
```



What are the thinks wrong with this?

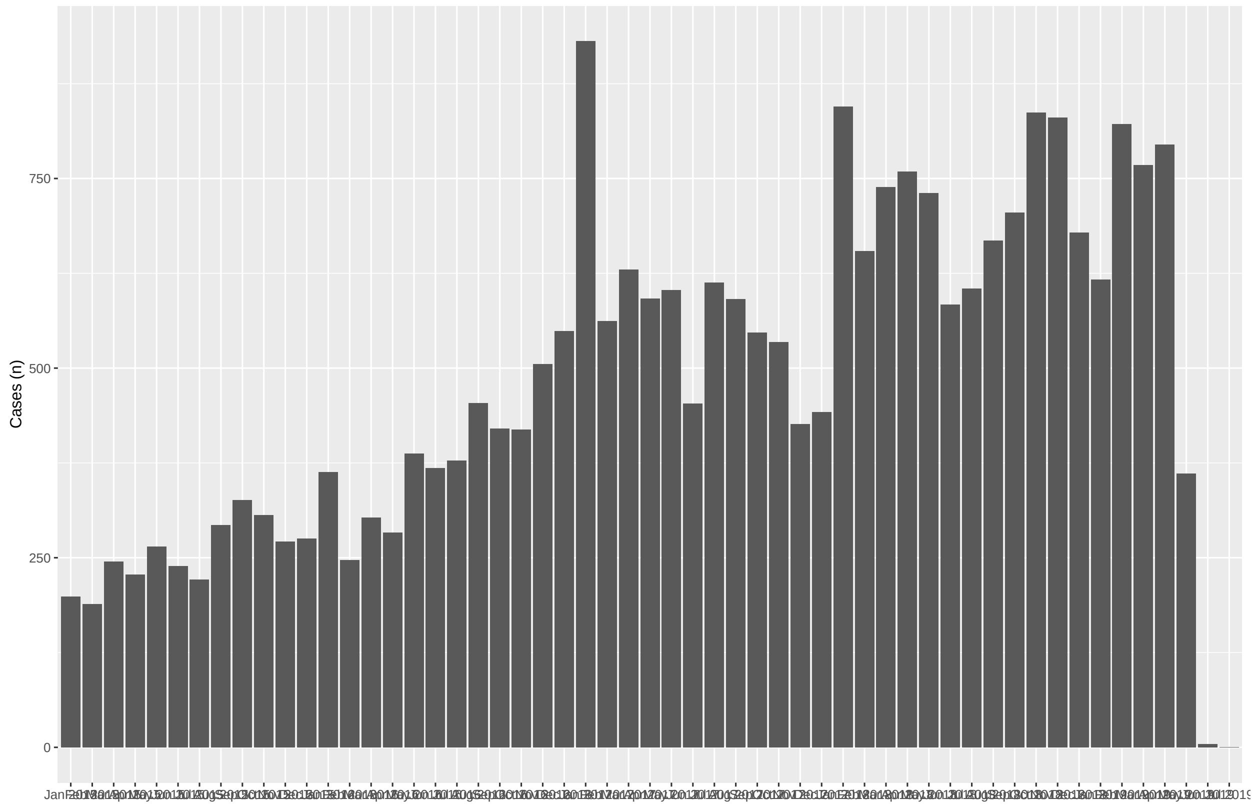
Firstly, the order of the months is not in calendar order but alphabetical so we can reorder how they are plotted by using `levels` in `factor`.

```
x$month <- factor(x$month, ordered = TRUE)
x$month_year <- factor(x$month_year, levels = x$month_year, ordered = TRUE)

plot0 <-
  ggplot(x, aes(x = month_year, y = cases)) +
  geom_bar(stat = "identity")

plot0
```



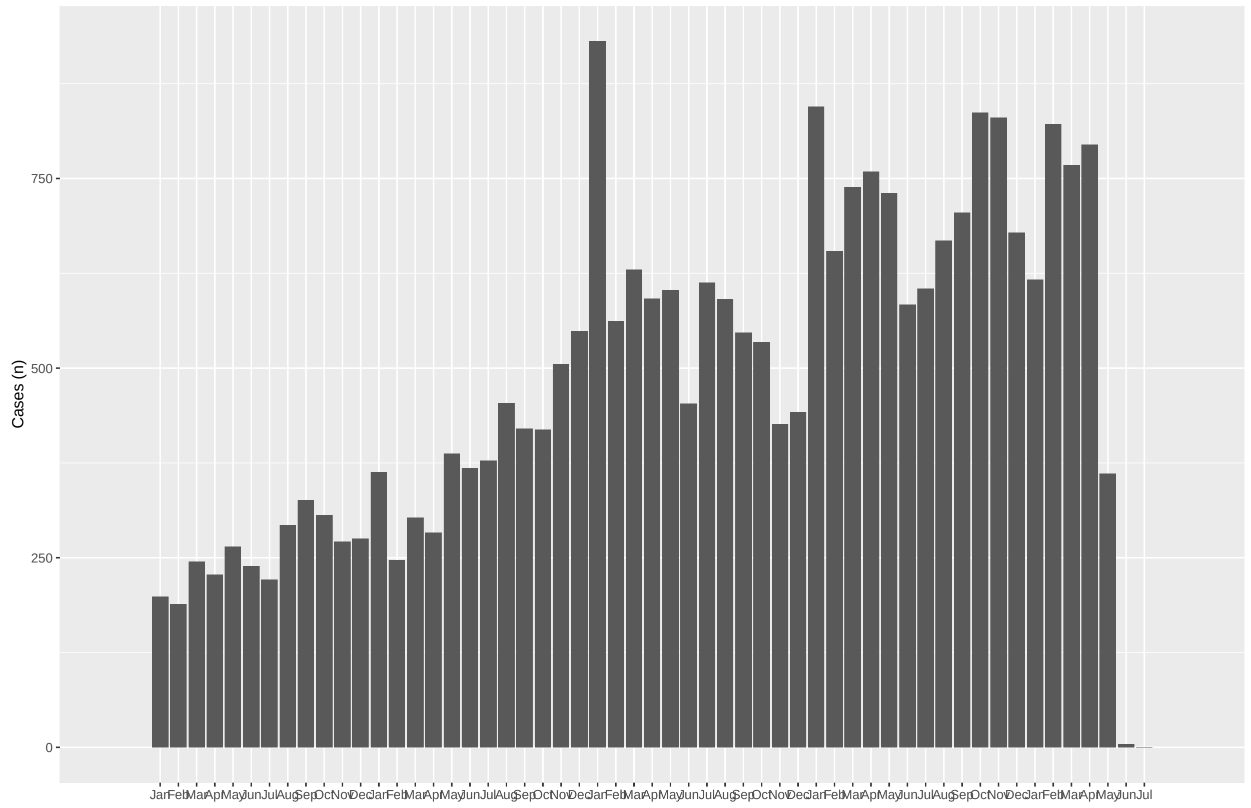
Lets rename the axis labels.

```
plot1 <-
  plot0 +
  xlab("") +
  ylab("Cases (n)")

plot1
```



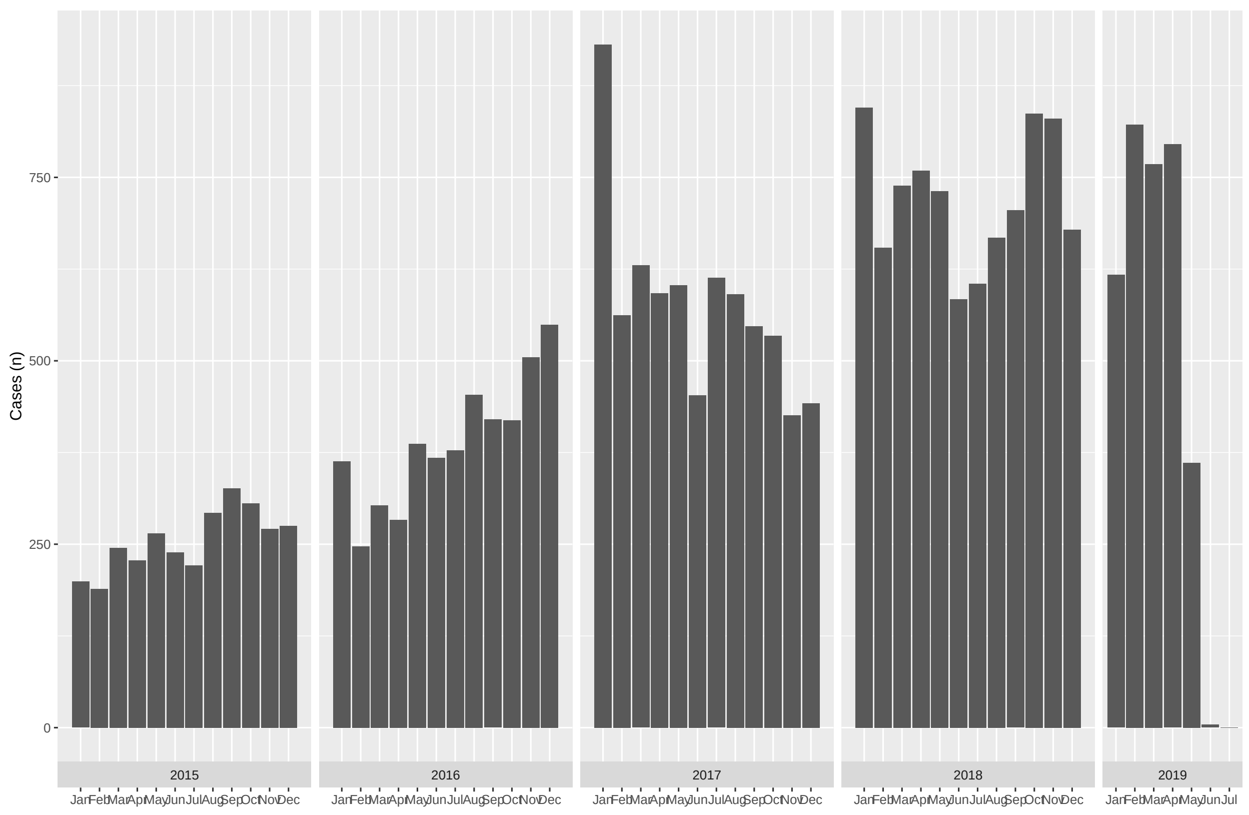
Now make some space around the edges of the plot

```r
plot2 <-
  plot1 +
  scale_x_discrete(breaks = x$month_year, labels = x$month,
                   expand = c(0.1,0.1))
plot2
```



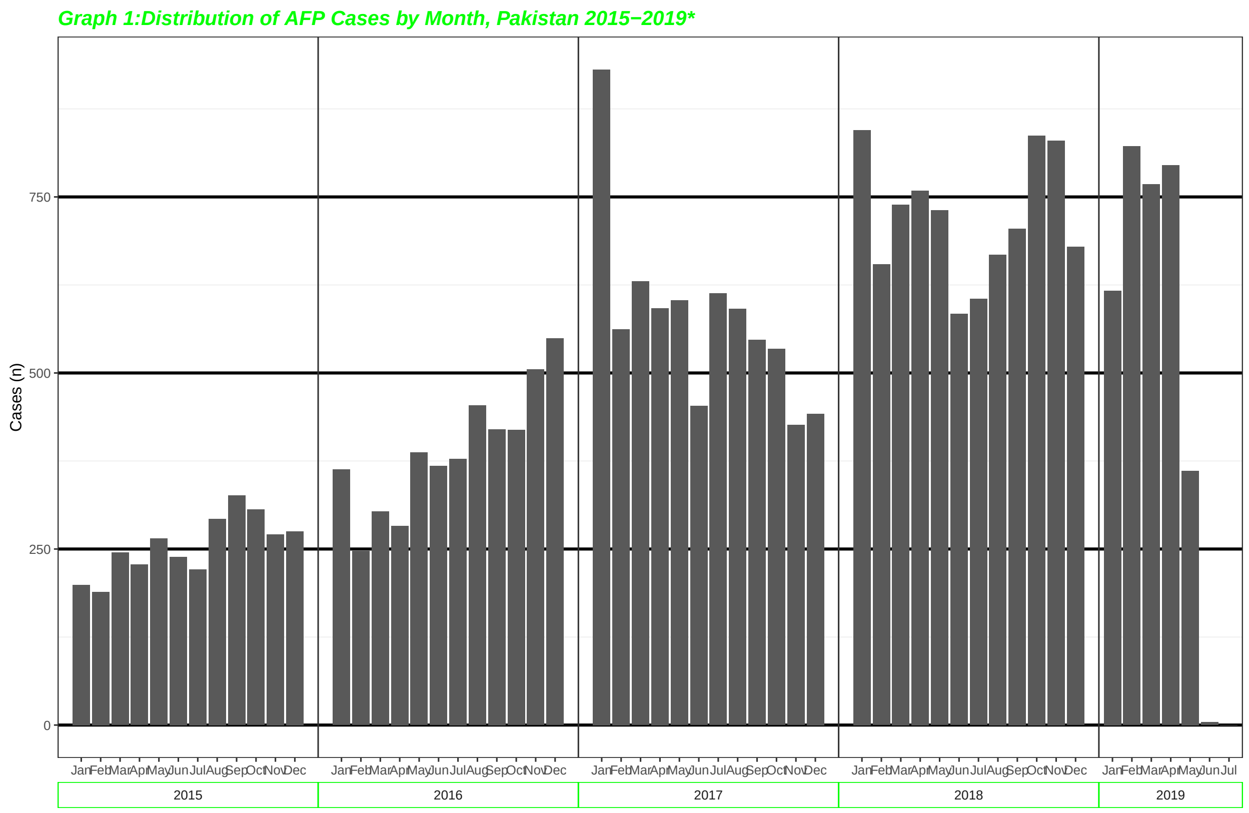
We want to make it clear which months are from which year by adding some space between years and labelling.

```
plot3 <-
  plot2 +
  facet_grid(~ YRONSET, space = "free_x", scales = "free_x", switch = "x")

plot3
```



Now we have the basic layout of the plot, we want to edit the style. From the example we were given there are several small asthetic changes we can make by using `theme()`.

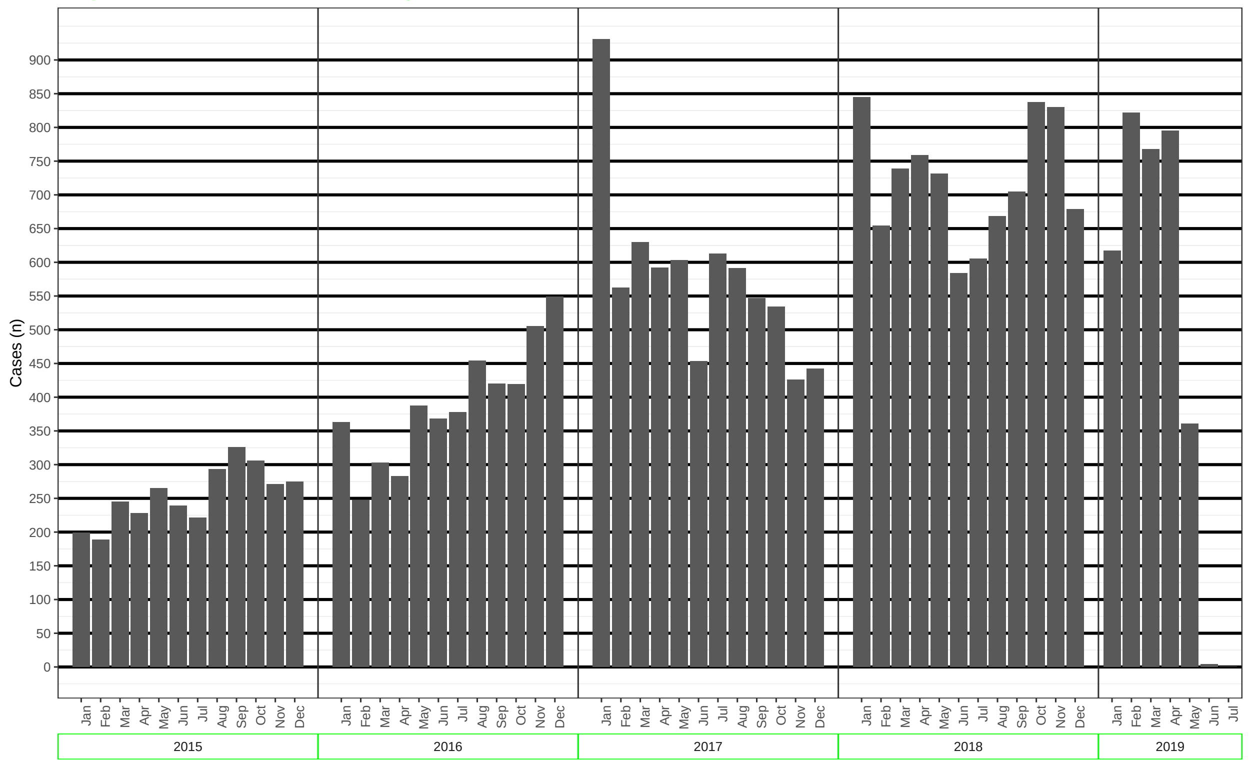
```
plot4 <-
  plot3 +
  theme_bw() +                                                                        # use a black and w
  ggtitle("Graph 1:Distribution of AFP Cases by Month, Pakistan 2015-2019*") +
  theme(strip.placement = "outside",                                            # swap the year and months labe
        strip.background = element_rect(fill = NA, colour = "green"),    # change background & border co
        panel.spacing = unit(0,"cm"),                                          # reduce the spacing between yea
        panel.grid.major.x = element_line(colour = NA, size = NULL, linetype = NULL,  # remove x and y
                                          lineend = NULL, color = NULL, arrow = NULL,
                                          inherit.blank = FALSE),
        panel.grid.major.y = element_line(colour = "black", size = 1, linetype = NULL,
                                          lineend = NULL, color = NULL, arrow = NULL,
                                          inherit.blank = FALSE),
        plot.title = element_text(color = "green", size = 14, face = "bold.italic")) # colour the title

plot4
```



Finally, we will do some finishing touches.

```
plot5 <-
  plot4 +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +              # rotate the x-axis months la
  scale_y_continuous(breaks = round(seq(min(x$cases), max(x$cases), by = 50), 1)) + # add horizontal bo
  labs(caption = "* Afp.rec Data as of  15-07-2019") +                               # add a footnote
  theme(
    plot.caption = element_text(size = 7, face = "italic"))                          # resize to small f

plot5
```

Graph 1:Distribution of AFP Cases by Month, Pakistan 2015−2019*



* Afp.rec Data as of 15−07−2019

## Environmental Sampling Results 2015-19 Pakistan

Read in the packages we will need.

```
library(dplyr)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 3.5.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.5.3
```

```
library(ggplot2)
library(scales)
```

```
## Warning: package 'scales' was built under R version 3.5.3
```

```
library(dataPakistan)
```

Load in the data `List of Env Samples 2015-2019.xlsx`.

```
file_name <- system.file(package = "dataPakistan", "extdata", "List of Env Samples 2015-2019.xlsx")

# dat <- read_excel(file.choose())
dat <-
  readxl::read_xlsx(
    path = file_name,
    range = "A2:F58",
    sheet = "Sheet1") #, sheet = 1)
```

If we looks the `YRONSET` variable we see that only the first month in a year has a value and the rest are empty (`NA`). We can fill in the empty months by carrying forward the previous value.

```
##############
# preprocess #
##############

# fill in the empty year of onset
# last observation carried forward (LOCF)
dat$YRONSET <- zoo::na.locf(dat$YRONSET)

# check
dat$YRONSET
```

```
##  [1] "2015"        "2015"        "2015"        "2015"        "2015"
##  [6] "2015"        "2015"        "2015"        "2015"        "2015"
## [11] "2015"        "2015"        "2016"        "2016"        "2016"
## [16] "2016"        "2016"        "2016"        "2016"        "2016"
## [21] "2016"        "2016"        "2016"        "2016"        "2017"
## [26] "2017"        "2017"        "2017"        "2017"        "2017"
## [31] "2017"        "2017"        "2017"        "2017"        "2017"
## [36] "2017"        "2018"        "2018"        "2018"        "2018"
## [41] "2018"        "2018"        "2018"        "2018"        "2018"
## [46] "2018"        "2018"        "2018"        "2019"        "2019"
## [51] "2019"        "2019"        "2019"        "2019"        "2019"
## [56] "Grand Total"
```

FOr each row (`rowwise`) we want to create a new value that tells us whether the sum total of the Positive, Negative and Under Process columns is the same as the Grand Total column. Call this new TRUE or FALSE values `check_total`.

To do this we can check for equivalence using the `==`.

```
dat <-
  dat %>%
  rowwise() %>%
  mutate(check_total = sum(Positive, Negative, `Under Process`, na.rm = TRUE) == `Grand Total`)
```

Now check to see if any of the rows do not have these things equal.

```
any(!dat$check_total)
```

```
## [1] FALSE
```

Looks ok! To keep things tidy lets remove the columns we won't need and remove the last row which is a grand total.

```
dat <- dat %>% select(-"Grand Total", -check_total)
dat <- dat[-nrow(dat), ]
```

Create a new combined month and year of onset column and make sure the order is in calendar time.

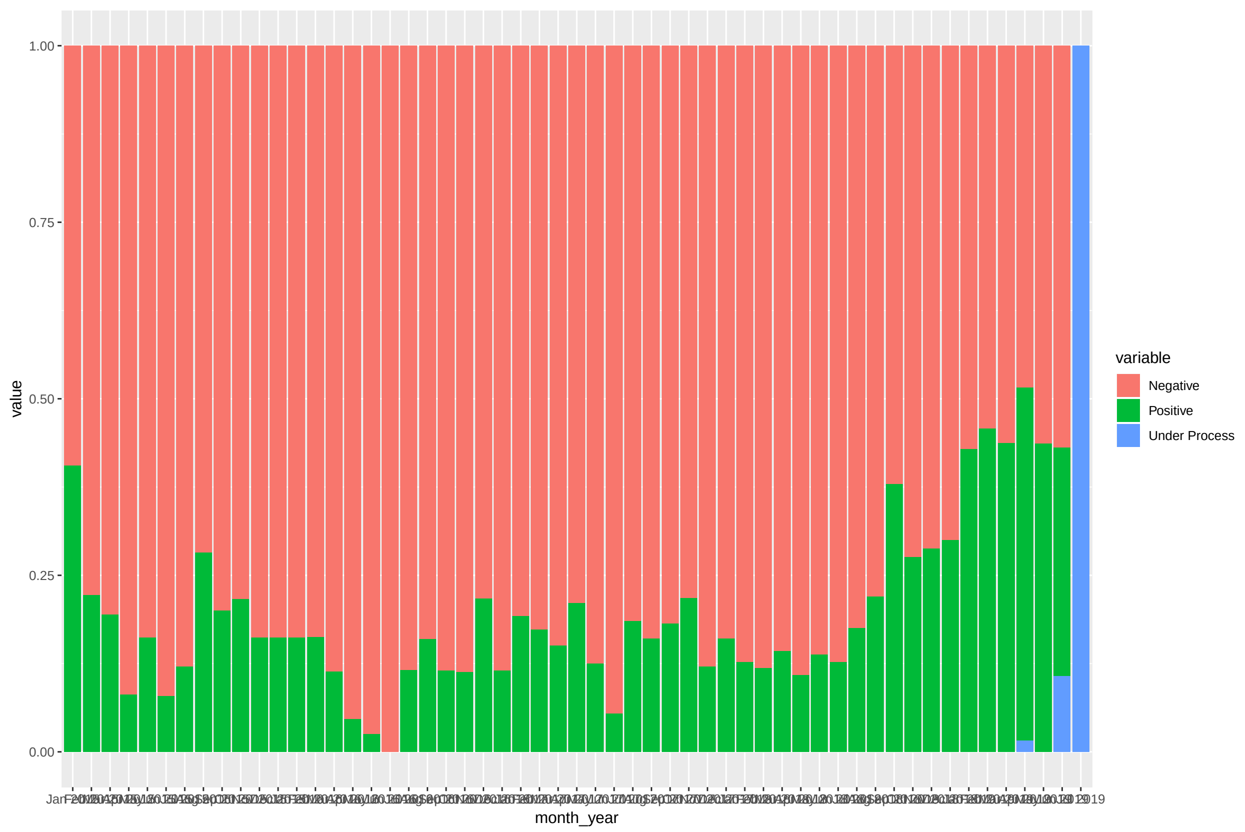
```r
dat$month_year <- paste(dat$MONTH, dat$YRONSET)
# dat$month <- factor(dat$month, ordered = TRUE)
dat$month_year <- factor(dat$month_year, levels = dat$month_year, ordered = TRUE)
```

Finally, we need to reshape the data to the format that `ggplot2` is expecting. To do this we will `melt` the data frame which means to transform it to a long array, where there is one column for the variable (`Positive`, `Negative`, `Under Process`) and one column for the corresponding value.

```r
library(reshape2)

x <- melt(dat, id.vars = c("YRONSET", "MONTH", "month_year"))
x$variable <- factor(x$variable,
                     levels = c("Positive", "Negative", "Under Process"))

# put Negative first in the ordering
x$variable <- relevel(x$variable, 'Negative')
```

```r
#########
# plots #
#########


# basic stacked bar plot
plot0 <-
  ggplot(x, aes(x = month_year, y = value, fill = variable)) +
  geom_bar(position = "fill", stat = "identity")

plot0
```

```
## Warning: Removed 55 rows containing missing values (position_stack).
```



```
library(scales)

plot1 <-
  plot0 +
  scale_fill_manual("legend",
                    values = c("Negative" = "lightgreen",         # change the colour scheme
                               "Positive" = "red",
                               "Under Process" = "grey")) +
  scale_y_continuous(labels = percent_format(),
                     breaks = seq(from = 0, to = 1, by = 0.1)) +  # change y-axis labels to %s
  scale_x_discrete(breaks = x$month_year, labels = x$MONTH,
                   expand = c(0.1,0.1)) +
  facet_grid(~ YRONSET, space = "free_x", scales = "free_x", switch = "x") +  # separate into years
  theme_bw() +                                                                # black and white theme
  theme(strip.placement = "outside",
        strip.background = element_rect(fill = NA, colour = "green"),     # swap x-axis labels, recolour
        panel.spacing = unit(0,"cm")) +
  ggtitle("Environmental Sampling Results 2015-19* \nPakistan") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

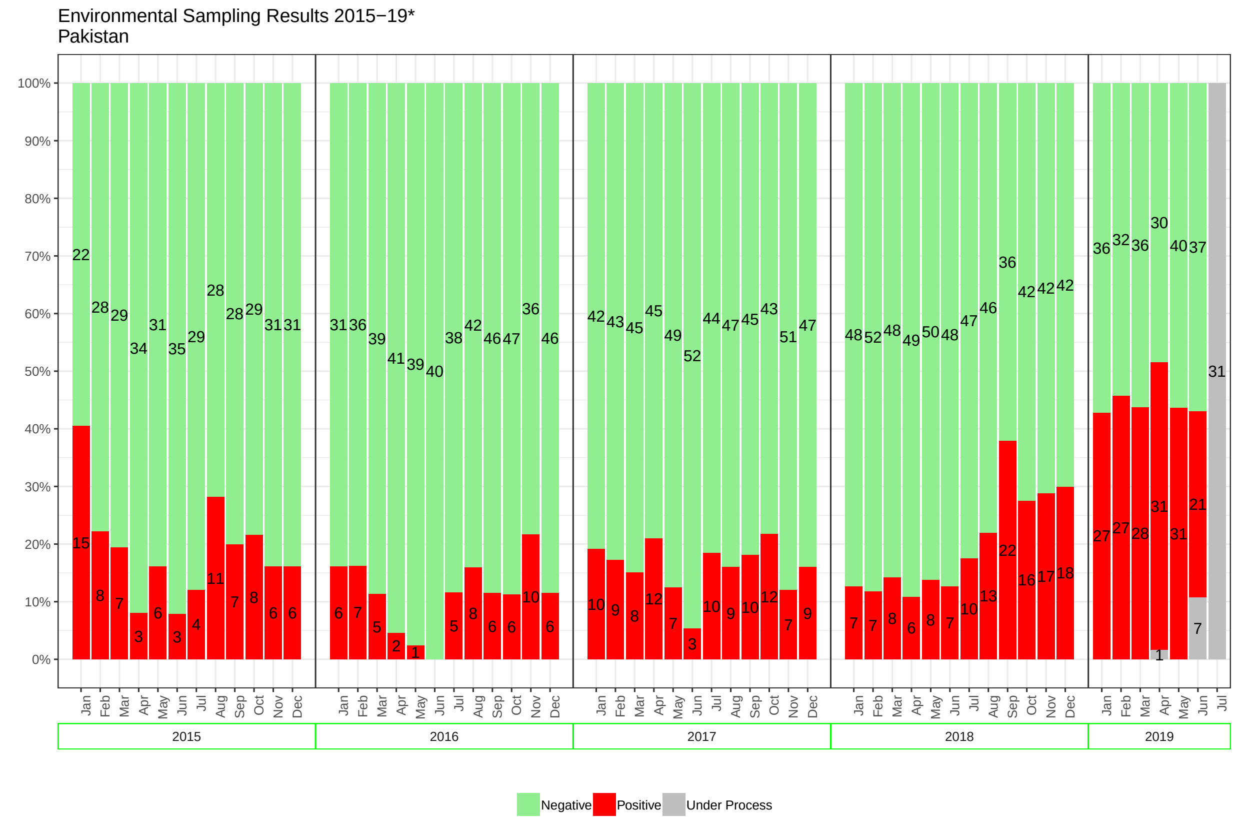
We want to move the legend to below the plot.

```
plot2 <-
  plot1 +
  xlab("") + ylab("") +
  theme(legend.position = "bottom", legend.title = element_blank())
```

Finally, annotate the bars with their values.

```
plot3 <-
  plot2 +
  geom_text(data = x, aes(y = value, label = value), position = position_fill(vjust = 0.5))

plot3
```

```
## Warning: Removed 55 rows containing missing values (position_stack).

## Warning: Removed 55 rows containing missing values (position_stack).
```



## MPQA _March SNID

Load packages.

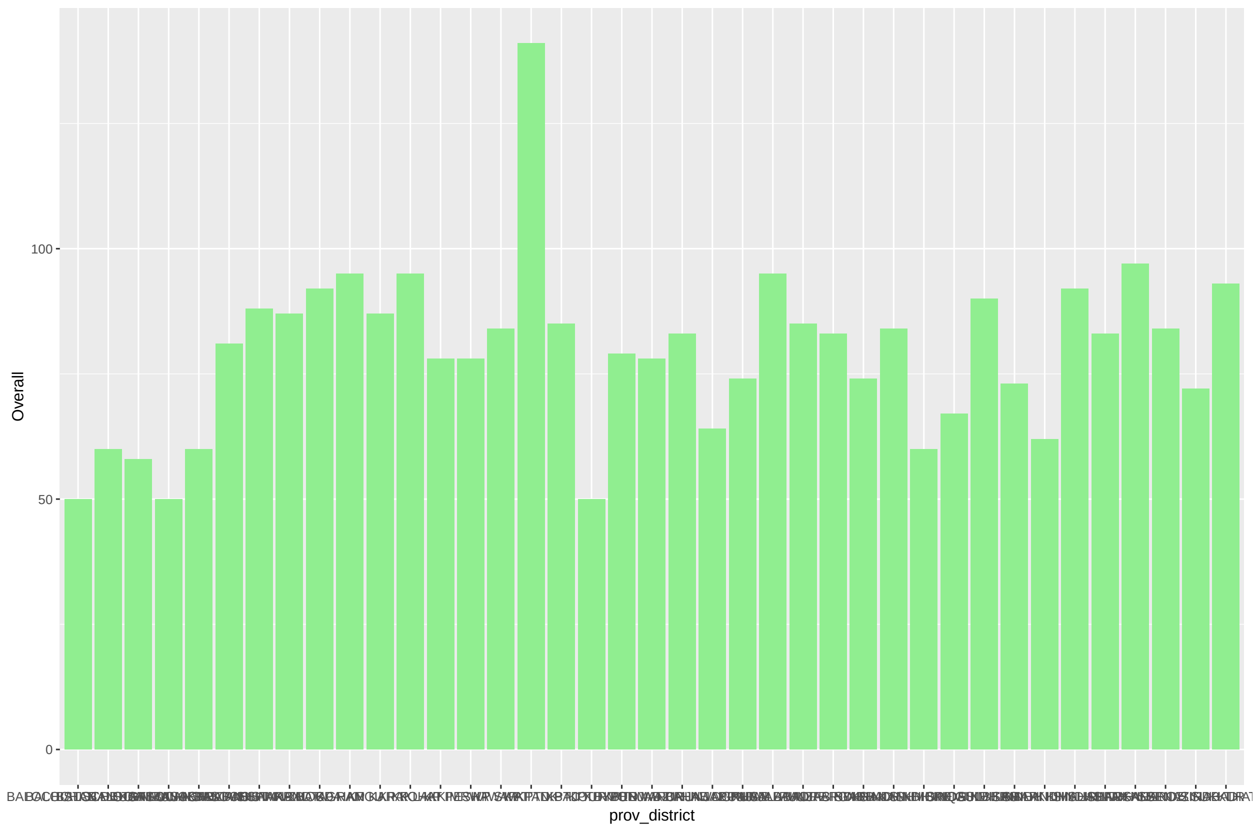
```
library(dplyr)
library(zoo)
library(reshape2)
library(ggplot2)
library(scales)

library(dataPakistan)
```

Load in data.

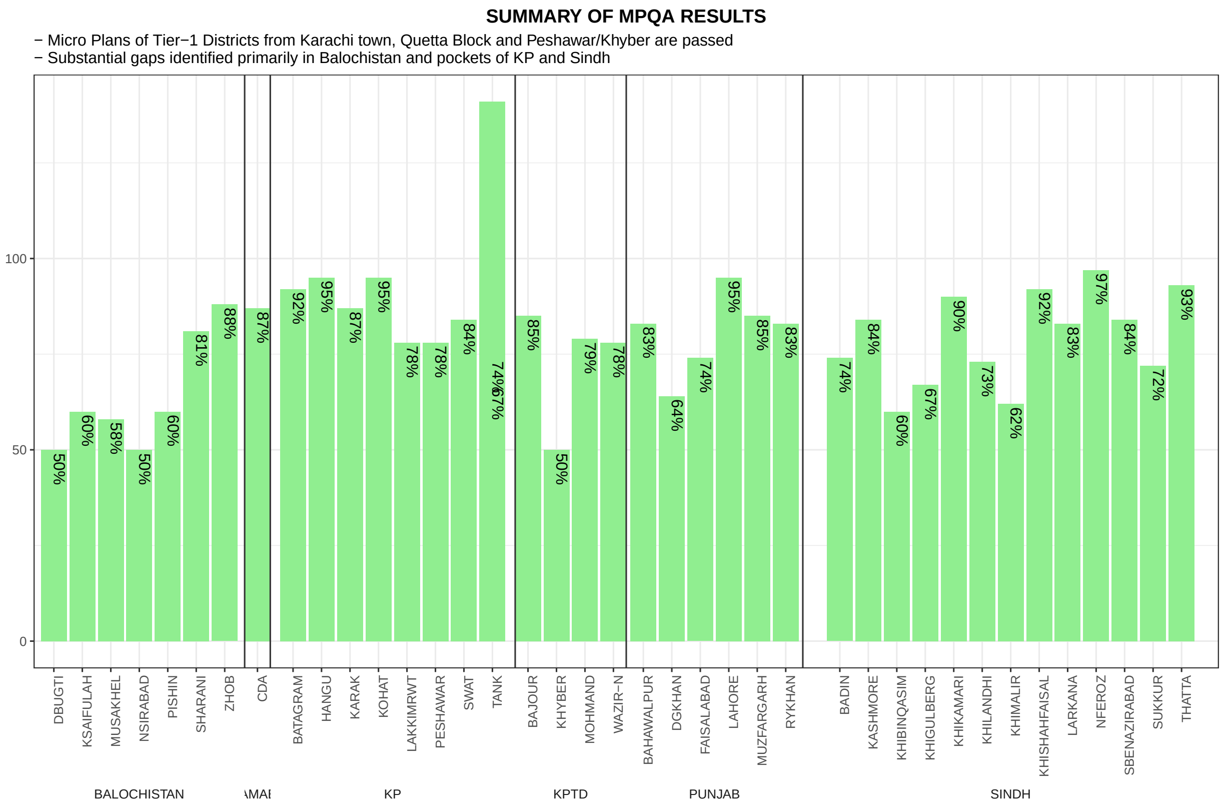
```
file_name <- system.file(package = "dataPakistan", "extdata", "MPQA _March SNID.xlsx")
dat <- readxl::read_xlsx(file_name)
```

```
##############
# preprocess #
##############

# remove columns we dont need
x <- dat %>% select(-DESK, -FIELD)

# create a Province-Ditrict single string
x$prov_district <- paste(x$"Province name", x$"District name")
```

```
#########
# plots #
#########

# basic plot
plot0 <-
  ggplot(x, aes(x = prov_district, y = Overall)) +
  geom_bar(stat = "identity", fill = "lightgreen")

plot0
```



```
plot1 <-
  plot0 +
  geom_text(aes(label = sprintf("%d%%", Overall)), vjust = 0, angle = -90, nudge_y = -5) +
  scale_x_discrete(breaks = x$prov_district, labels = x$"District name",
                   expand = c(0.1,0.1)) +
  facet_grid(~ `Province name`, space = "free_x", scales = "free_x", switch = "x") +
  theme_bw() +
  theme(strip.placement = "outside",
        strip.background = element_rect(fill = NA, linetype = 0),
        plot.title = element_text(hjust = 0.5, face = "bold"),
        panel.spacing = unit(0,"cm")) +
  labs(subtitle =
         "- Micro Plans of Tier-1 Districts from Karachi town, Quetta Block and Peshawar/Khyber are pas
  ggtitle("SUMMARY OF MPQA RESULTS", ) +
  xlab("") + ylab("") + theme(axis.text.x = element_text(angle = 90, hjust = 1))
  # geom_text(data = x, aes(y = value, label = value), position = position_fill(vjust = 0.5))

plot1
```


SUMMARY OF MPQA RESULTS
− Micro Plans of Tier−1 Districts from Karachi town, Quetta Block and Peshawar/Khyber are passed
− Substantial gaps identified primarily in Balochistan and pockets of KP and Sindh
DBUGTI
KSAIFULAH
MUSAKHEL
NSIRABAD
PISHIN
SHARANI
ZHOB
CDA
BATAGRAM
HANGU
KARAK
KOHAT
LAKKIMRWT
PESHAWAR
SWAT
TANK
BAJOUR
KHYBER
MOHMAND
WAZIR−N
BAHAWALPUR
DGKHAN
FAISALABAD
LAHORE
MUZFARGARH
RYKHAN
BADIN
KASHMORE
KHIBINQASIM
KHIGULBERG
KHIKAMARI
KHILANDHI
KHIMALIR
KHISHAHFAISAL
LARKANA
NFEROZ
SBENAZIRABAD
SUKKUR
THATTA
BALOCHISTAN
\MAI
KP
KPTD
PUNJAB
SINDH