R3

2025-06-28

R Markdown

```
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   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
#National Parks in California
ca <- read_csv("data/ca.csv")
```

```
## Rows: 789 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): region, state, code, park_name, type
## dbl (2): visitors, year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Acadia National Park
acadia <- read_csv("data/acadia.csv")
```

```
## Rows: 98 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): region, state, code, park_name, type
## dbl (2): visitors, year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Southeast US National Parks
se <- read_csv("data/se.csv")
```

```
## Rows: 453 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): region, state, code, park_name, type
## dbl (2): visitors, year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#2016 Visitation for all Pacific West National Parks
visit_16 <- read_csv("data/visit_16.csv")
```

```
## Rows: 17 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): region, state, code, park_name, type
## dbl (2): visitors, year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#All Nationally designated sites in Massachusetts
mass <- read_csv("data/mass.csv")
```

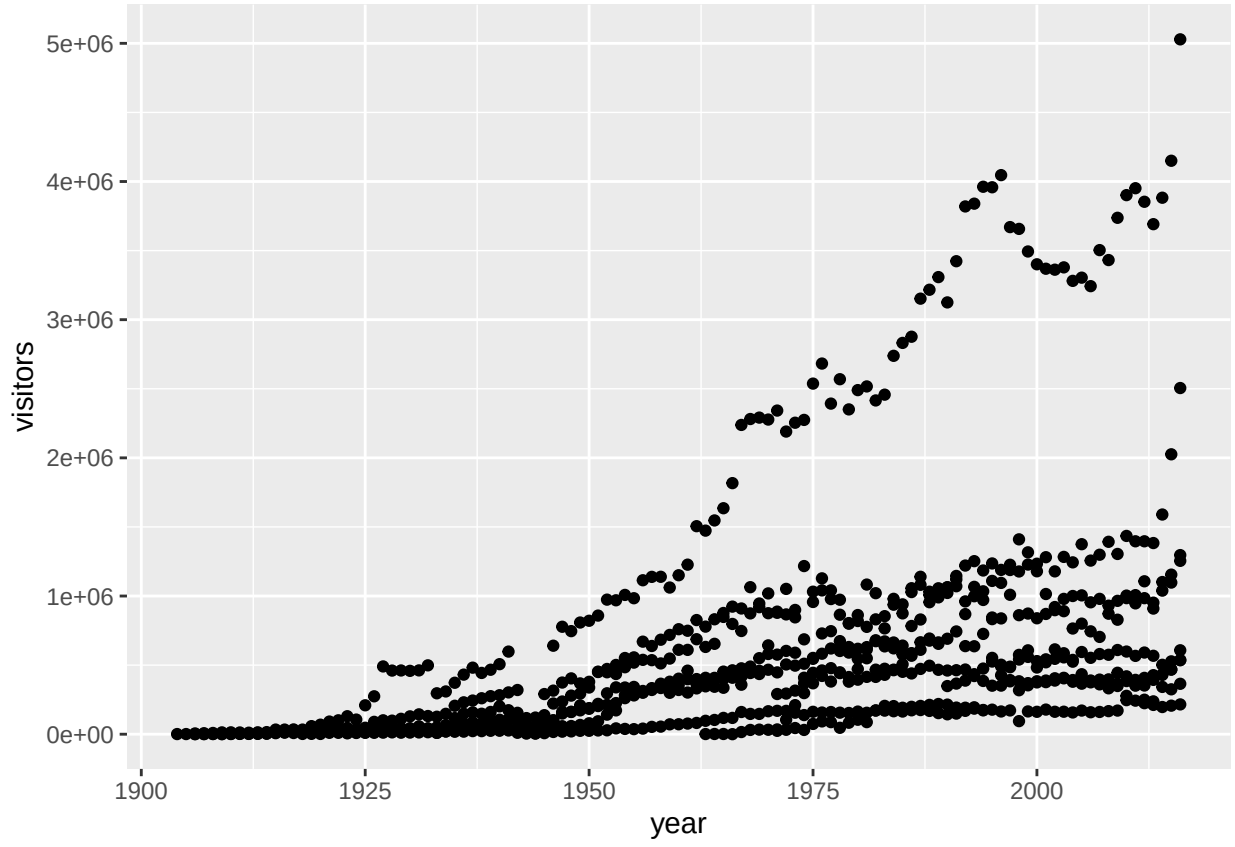
```
## Rows: 13 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): region, state, code, park_name, type
```

```
## dbl (2): visitors, year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

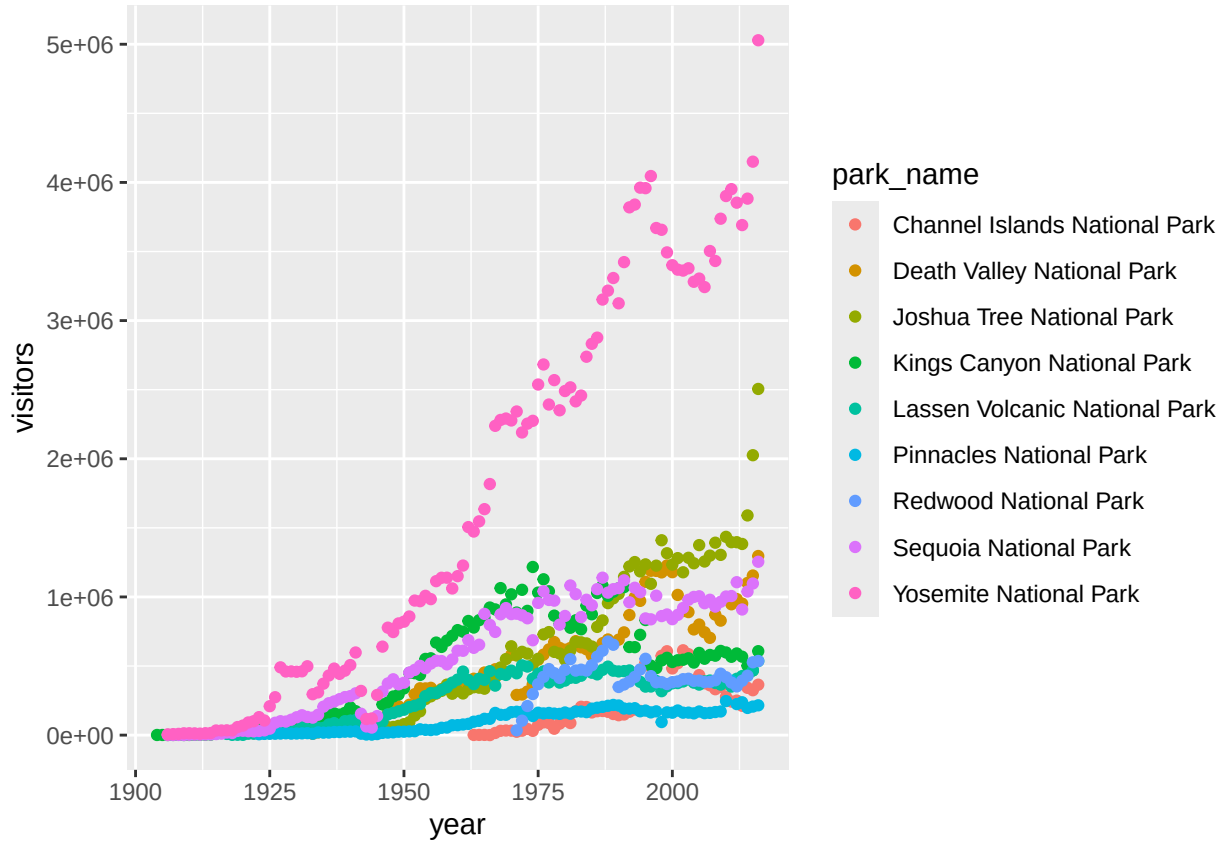
```
head(ca)
```

```
## # A tibble: 6 x 7
##   region state code  park_name                       type          visitors  year
##   <chr>  <chr> <chr> <chr>                           <chr>            <dbl> <dbl>
## 1 PW     CA    CHIS  Channel Islands National Park National Park      1200  1963
## 2 PW     CA    CHIS  Channel Islands National Park National Park      1500  1964
## 3 PW     CA    CHIS  Channel Islands National Park National Park      1600  1965
## 4 PW     CA    CHIS  Channel Islands National Park National Park       300  1966
## 5 PW     CA    CHIS  Channel Islands National Park National Park     15700  1967
## 6 PW     CA    CHIS  Channel Islands National Park National Park     31000  1968
```

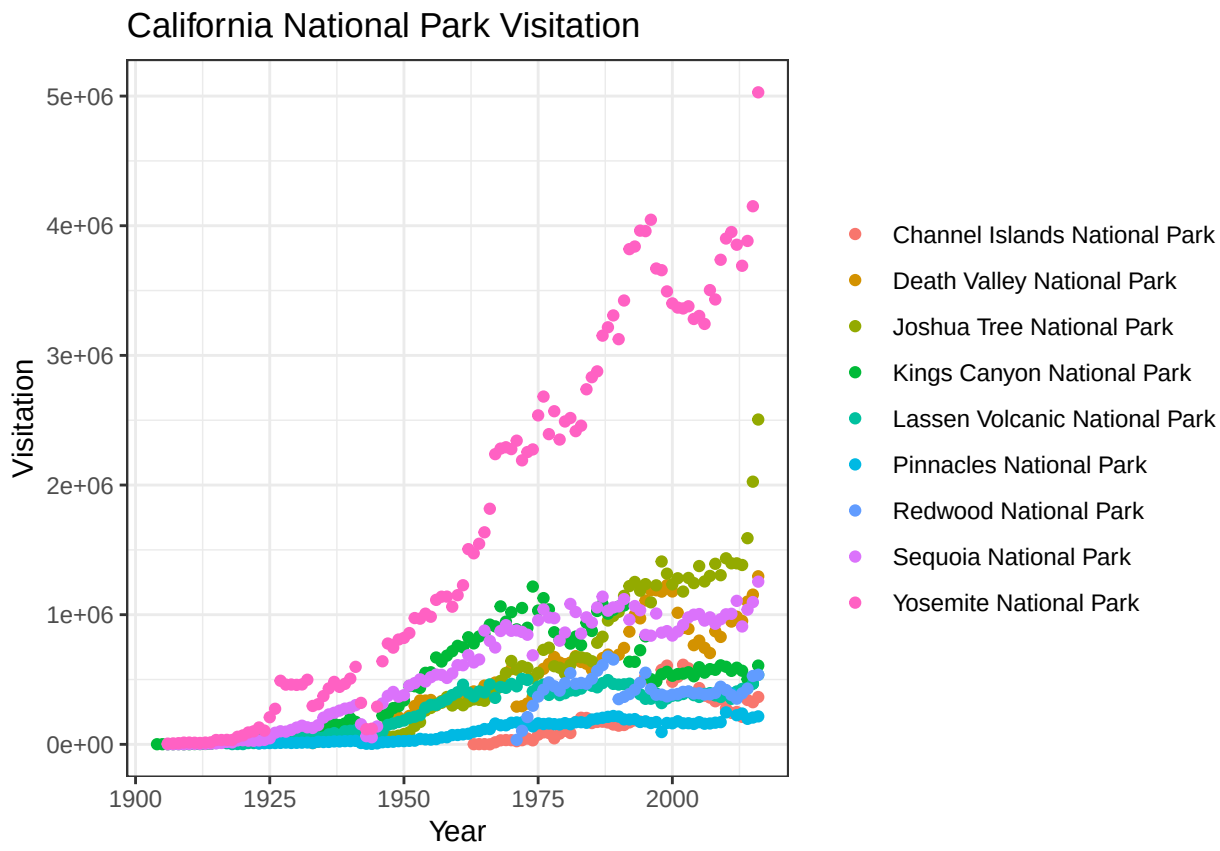
```
ggplot(data = ca) +
 geom_point(aes(x = year, y = visitors))
```



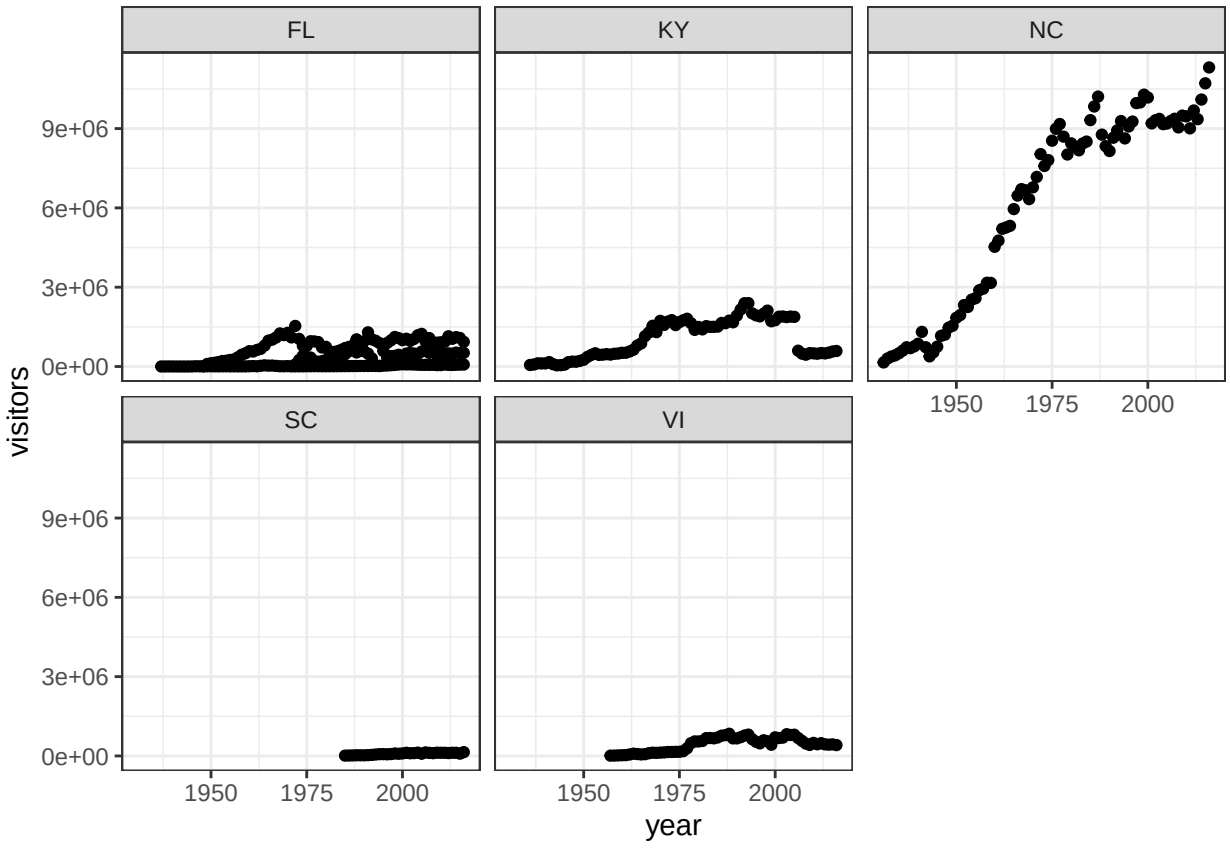
```
ggplot(data = ca) +
 geom_point(aes(x = year, y = visitors, color = park_name))
```



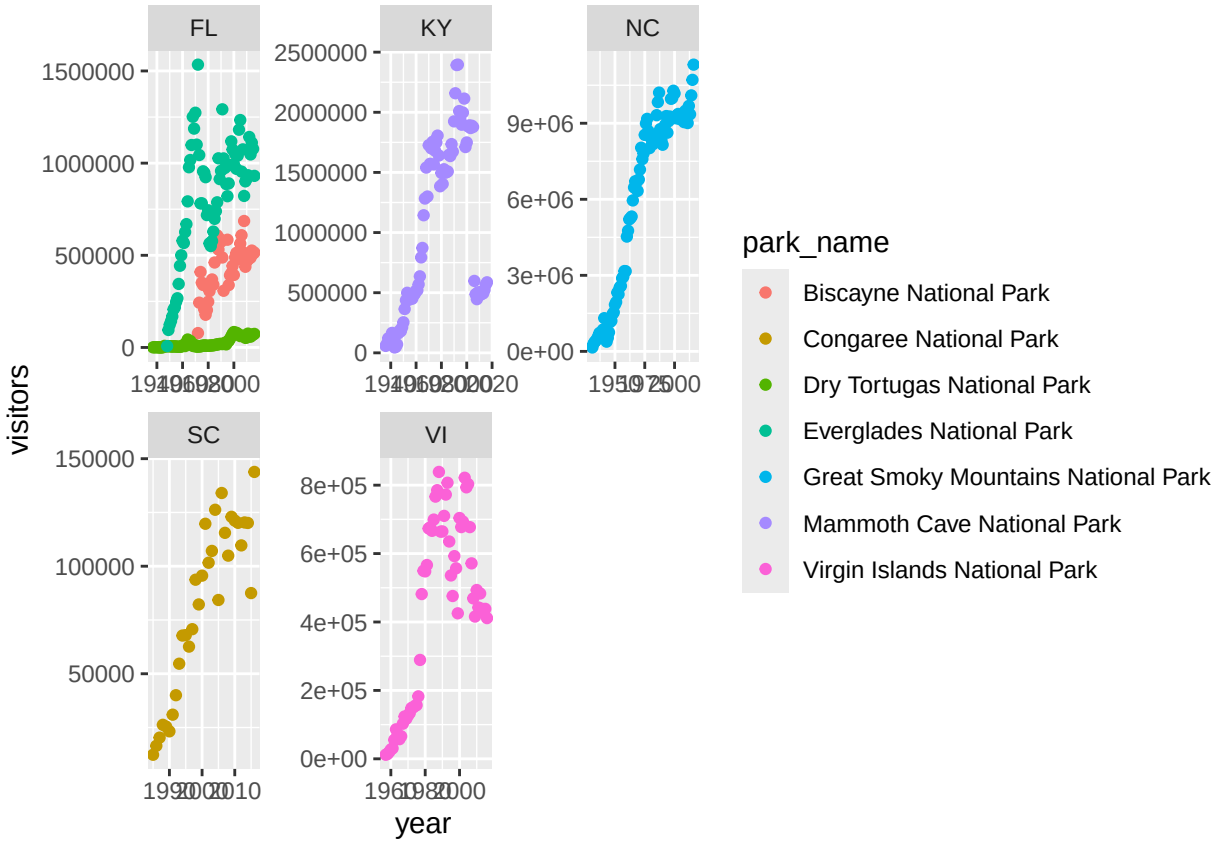

```
ggplot(data = ca) +
 geom_point(aes(x = year, y = visitors, color = park_name)) +
 labs(x = "Year",
 y = "Visitation",
 title = "California National Park Visitation") +
 theme_bw() +
 theme(legend.title=element_blank())
```



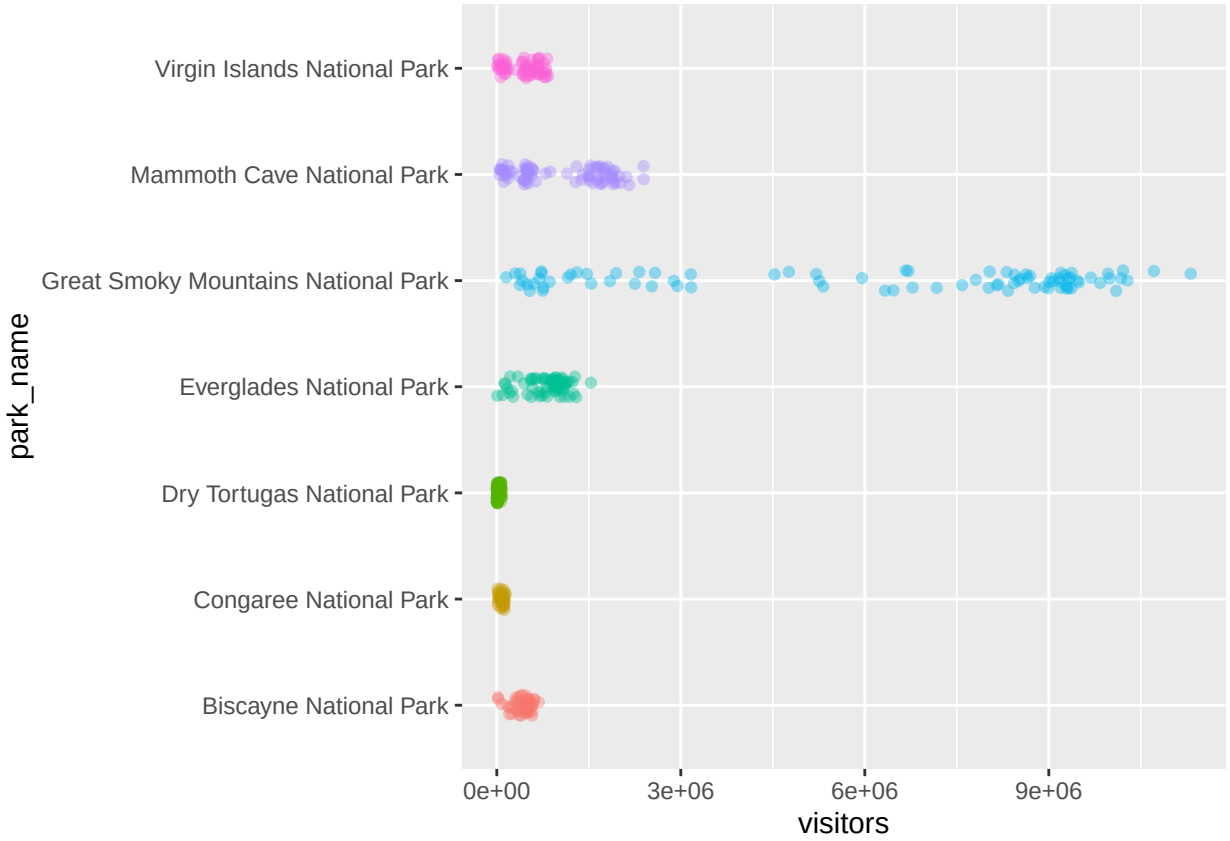

```
ggplot(data = se) +
 geom_point(aes(x = year, y = visitors)) +
 facet_wrap(~ state)+
  theme_bw()
```



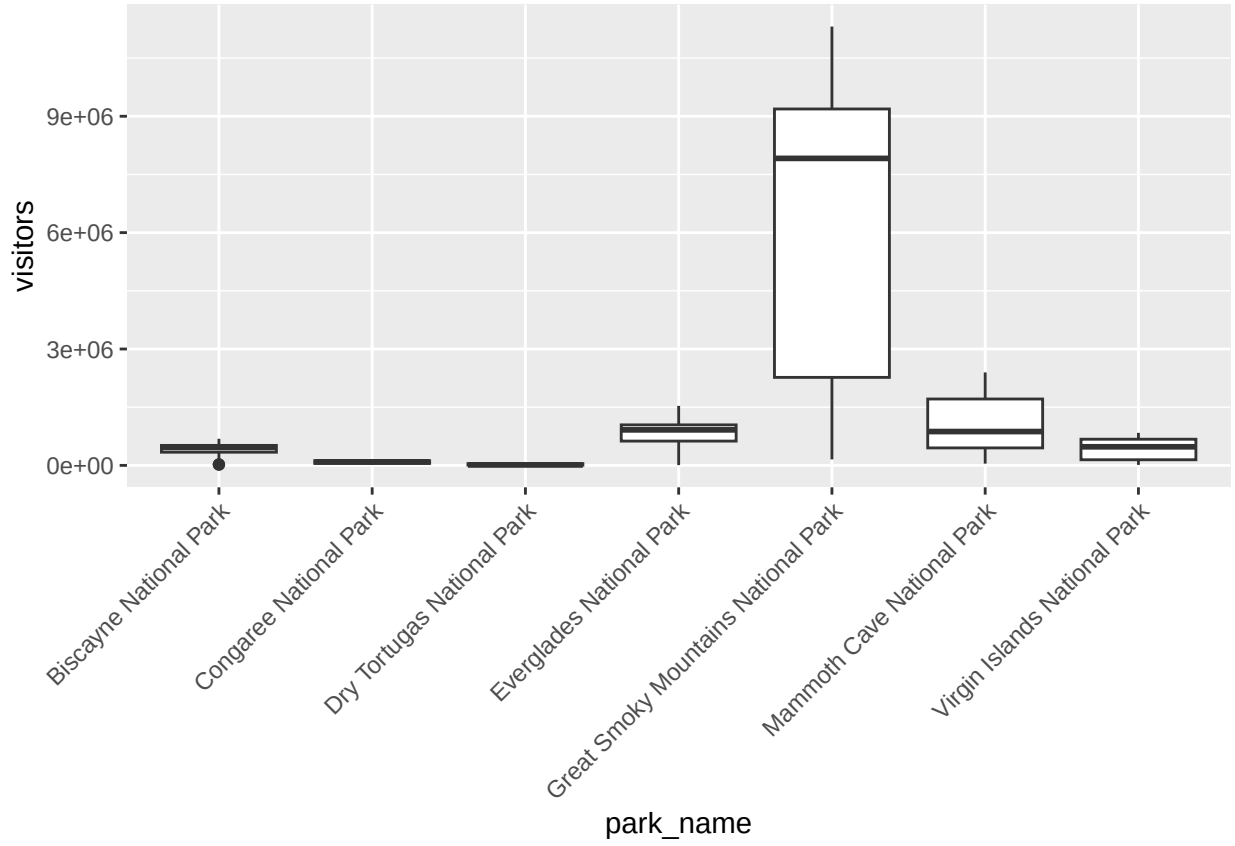

```
ggplot(data = se) +
 geom_point(aes(x = year, y = visitors, color = park_name)) +
 facet_wrap(~ state, scales = "free")
```



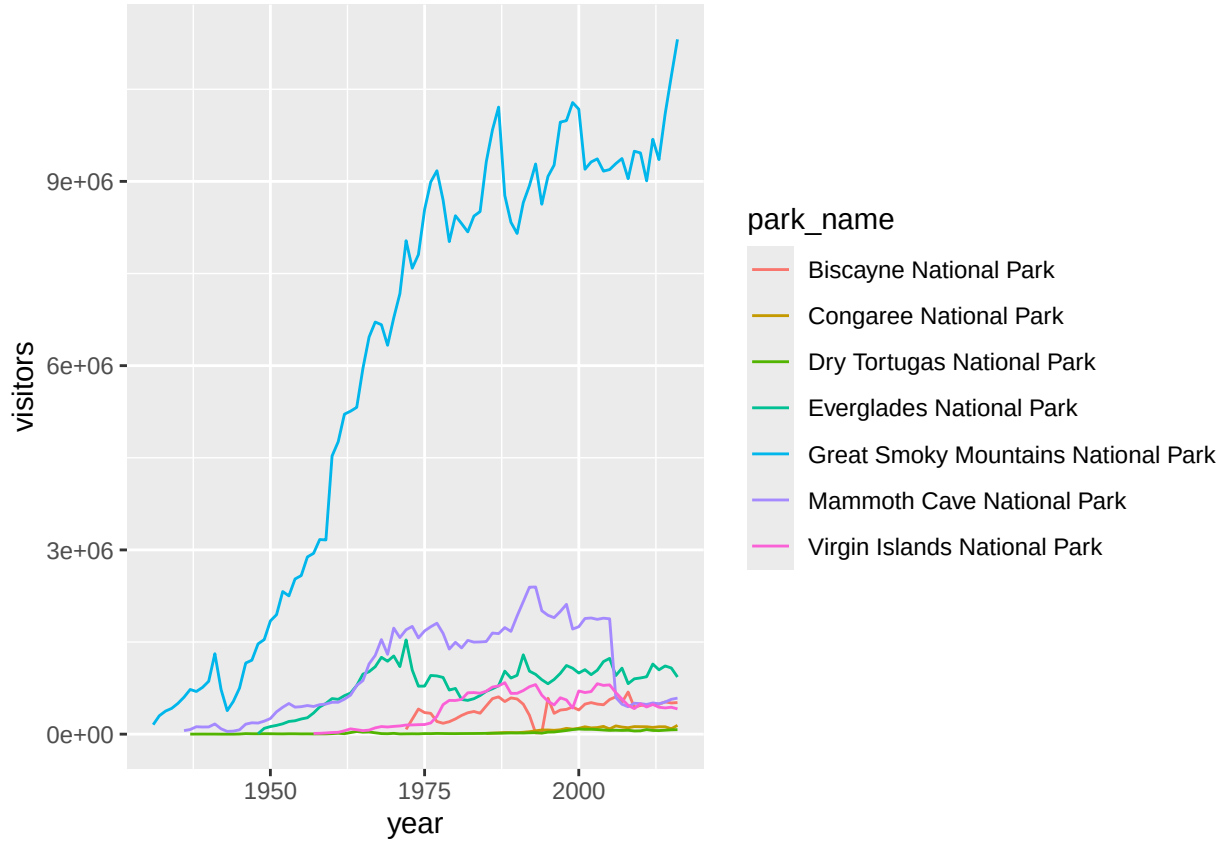

```
ggplot(data = se) +
 geom_jitter(aes(x = park_name, y = visitors, color = park_name),
 width = 0.1,
 alpha = 0.4) +
 coord_flip() +
 theme(legend.position = "none")
```




```
ggplot(se, aes(x = park_name, y = visitors)) +
 geom_boxplot() +
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

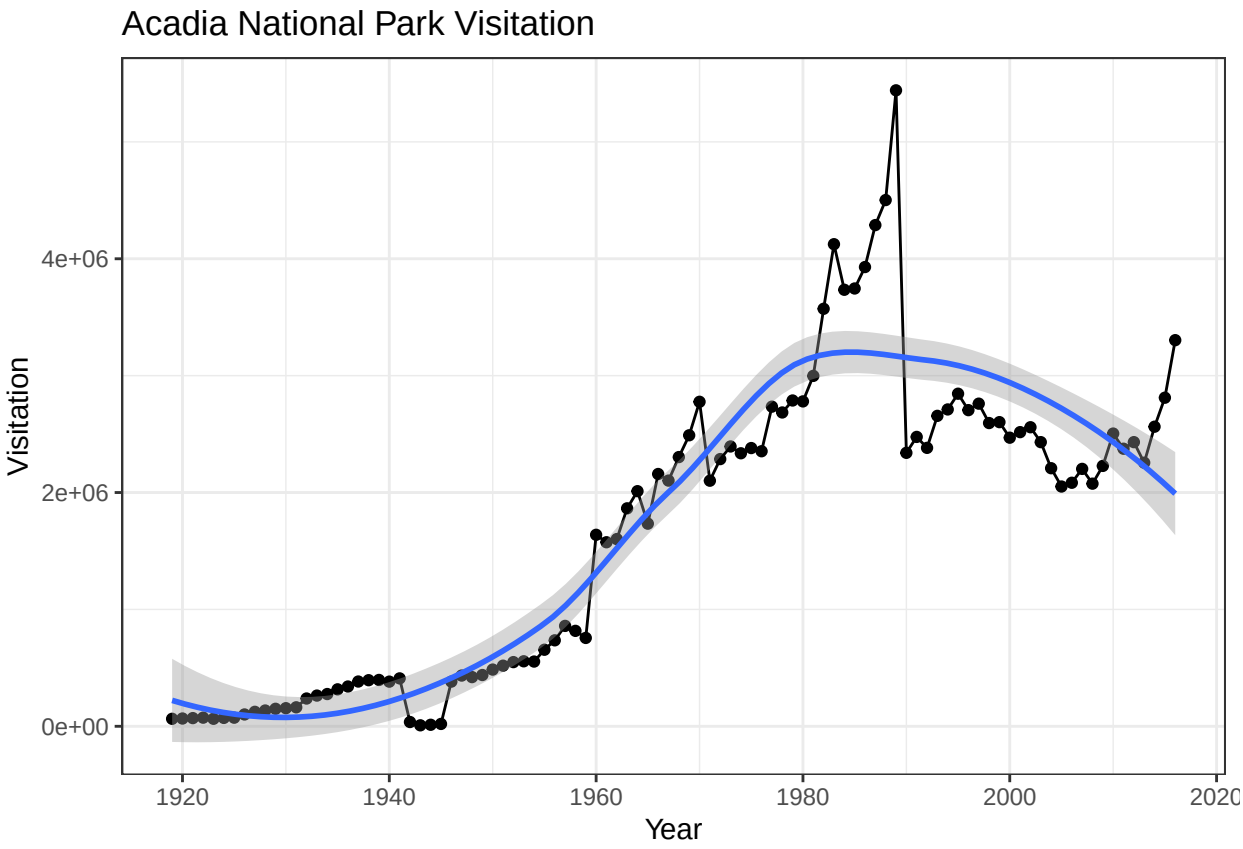



```
ggplot(se, aes(x = year, y = visitors, color = park_name)) +
 geom_line()
```

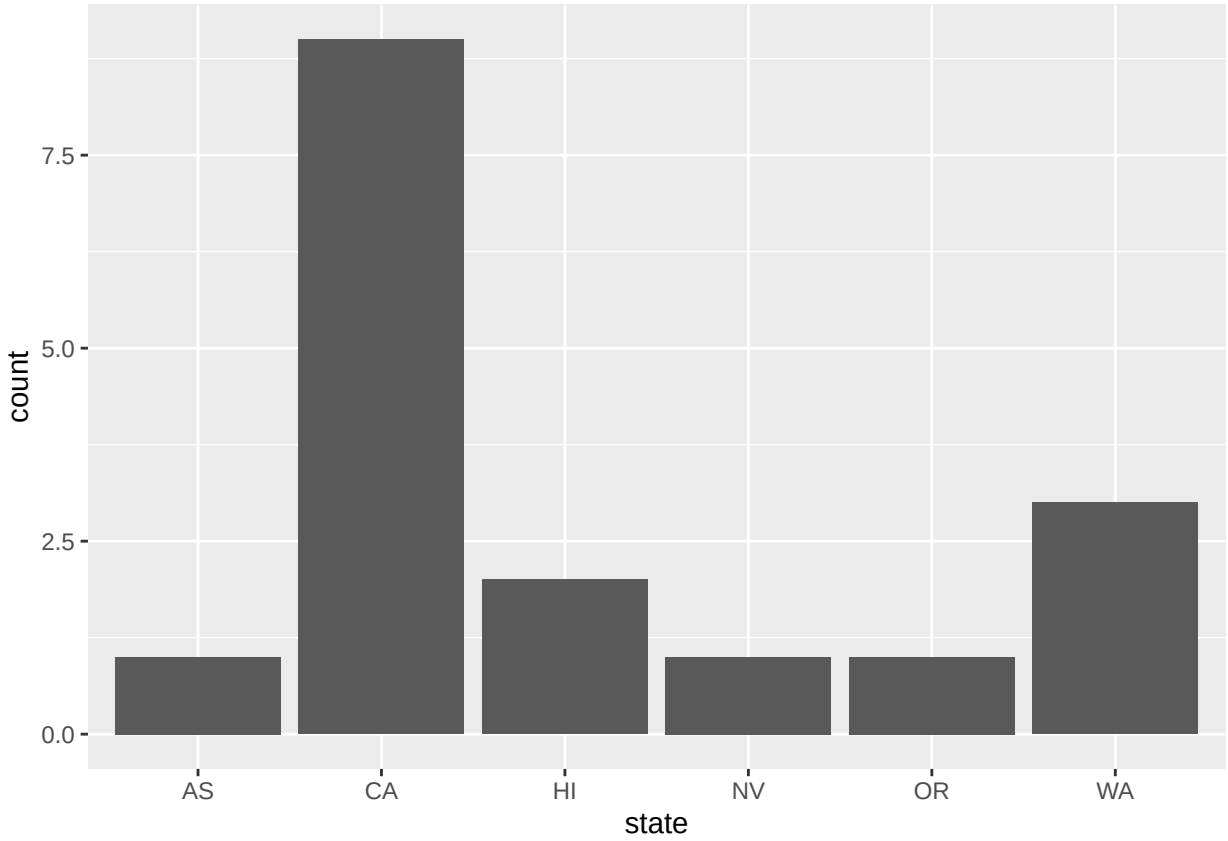



```
ggplot(data = acadia) +
 geom_point(aes(x = year, y = visitors)) +
 geom_line(aes(x = year, y = visitors)) +
 geom_smooth(aes(x = year, y = visitors)) +
 labs(title = "Acadia National Park Visitation",
 y = "Visitation",
 x = "Year") +
 theme_bw()
```

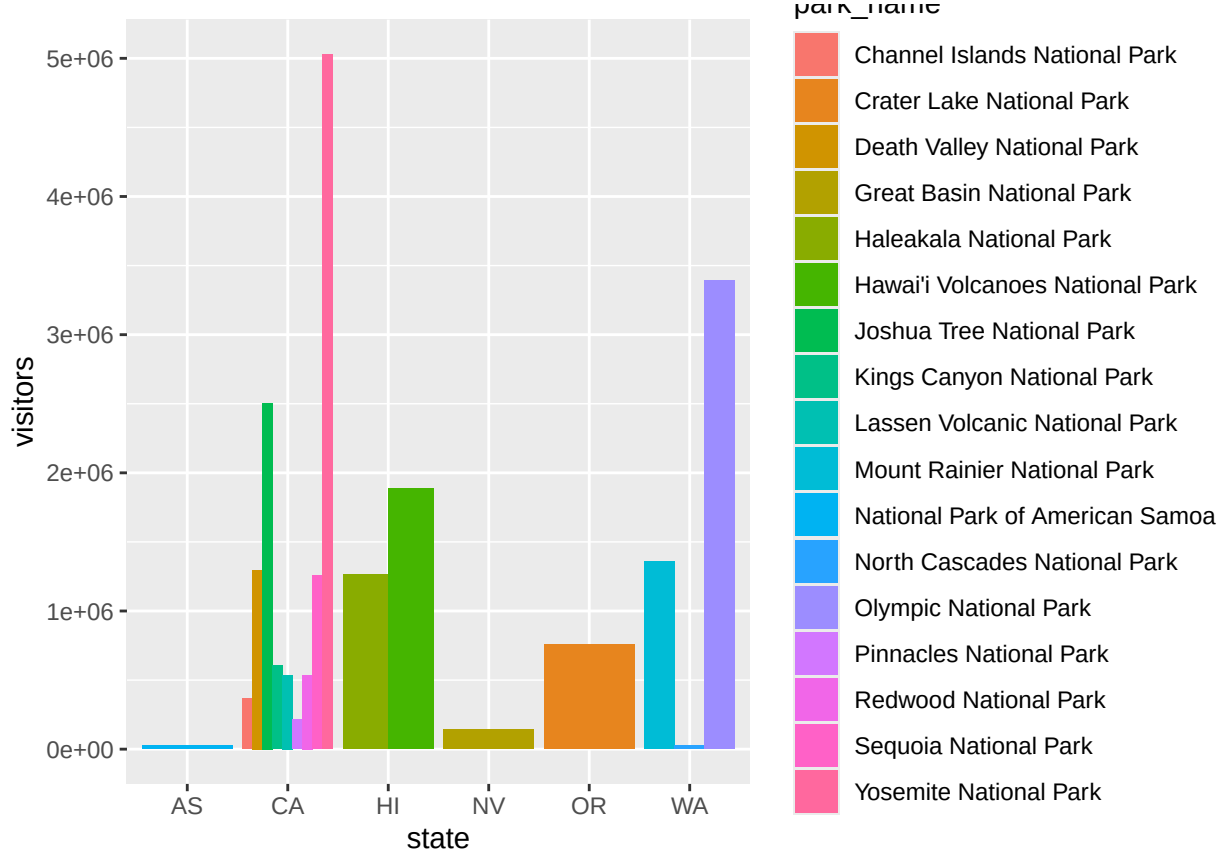
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
ggplot(data = visit_16, aes(x = state)) +
 geom_bar()
```



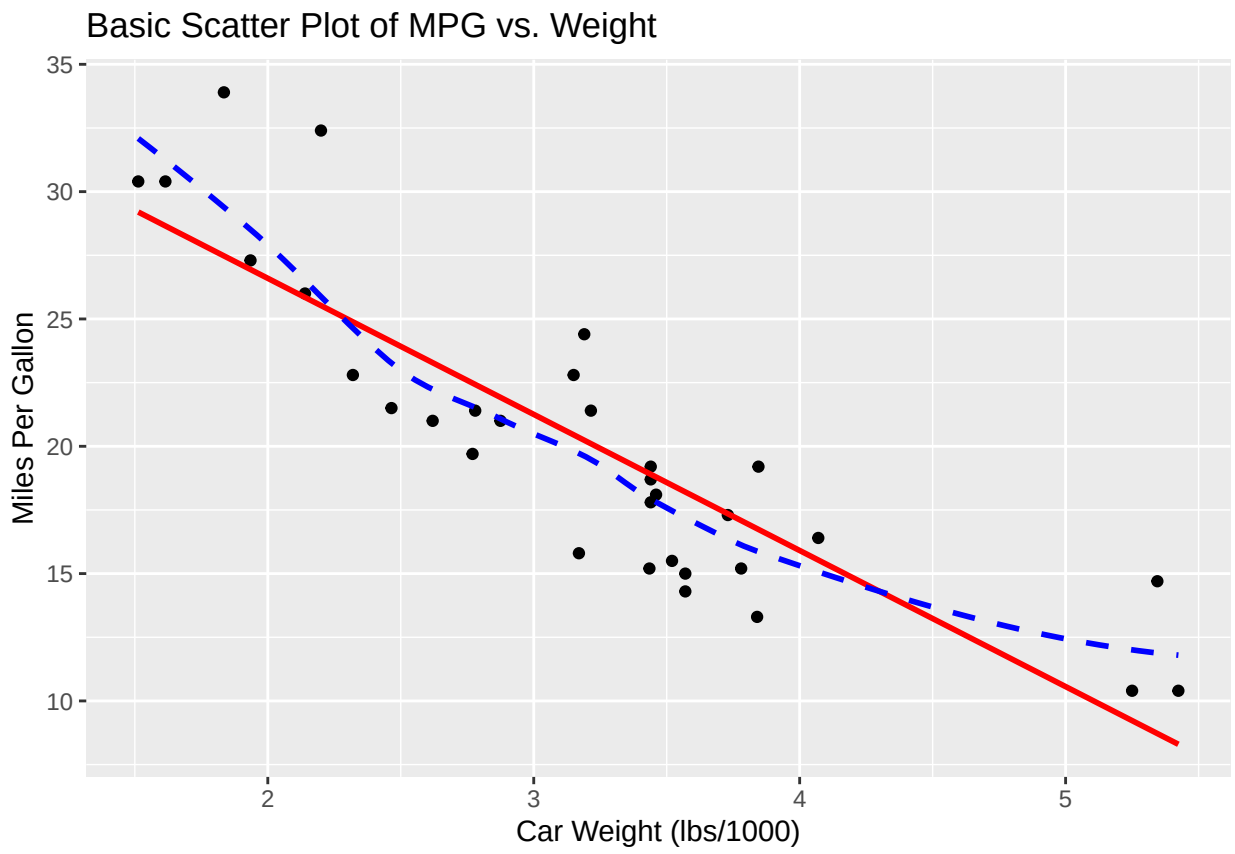

```
ggplot(data = visit_16, aes(x = state,y=visitors,fill = park_name)) +
 geom_bar(stat = "identity", position = "dodge")
```




```
my_plot <- ggplot(data = mass) +
 geom_bar(aes(x = type, fill = park_name)) +
 labs(x = "", y = "")+
 theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 7))
ggsave("name_of_file.png", my_plot, width = 15, height = 10)
```

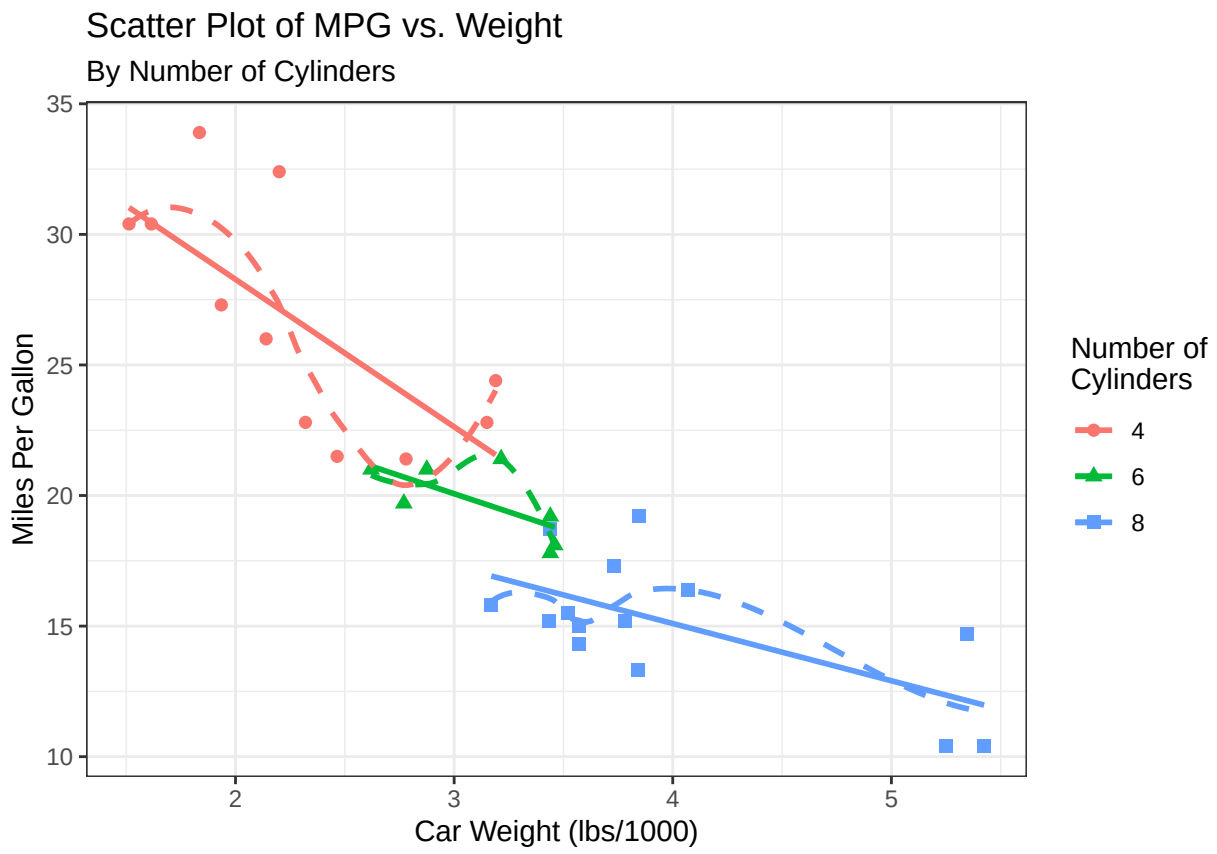
```
data(mtcars) #1
mtcars |> ggplot(aes(x=wt, y=mpg)) +
 geom_point() +
 geom_smooth(method="lm", se=FALSE, color="red") +
 geom_smooth(method="loess", se=FALSE,
 color="blue", linetype="dashed") +
 labs(title = "Basic Scatter Plot of MPG vs. Weight",
 x = "Car Weight (lbs/1000)", y = "Miles Per Gallon")
```

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




```
mtcars |> ggplot(aes(x=wt, y=mpg, color=factor(cyl),
 shape=factor(cyl))) +
 geom_point(size=2) +
 geom_smooth(method="lm", se=FALSE) +
 geom_smooth(method="loess", se=FALSE, linetype="dashed") +
 labs(title = "Scatter Plot of MPG vs. Weight",
 subtitle = "By Number of Cylinders",
 x = "Car Weight (lbs/1000)", y = "Miles Per Gallon",
 color = "Number of \nCylinders",
 shape = "Number of \nCylinders") +
 theme_bw()
```

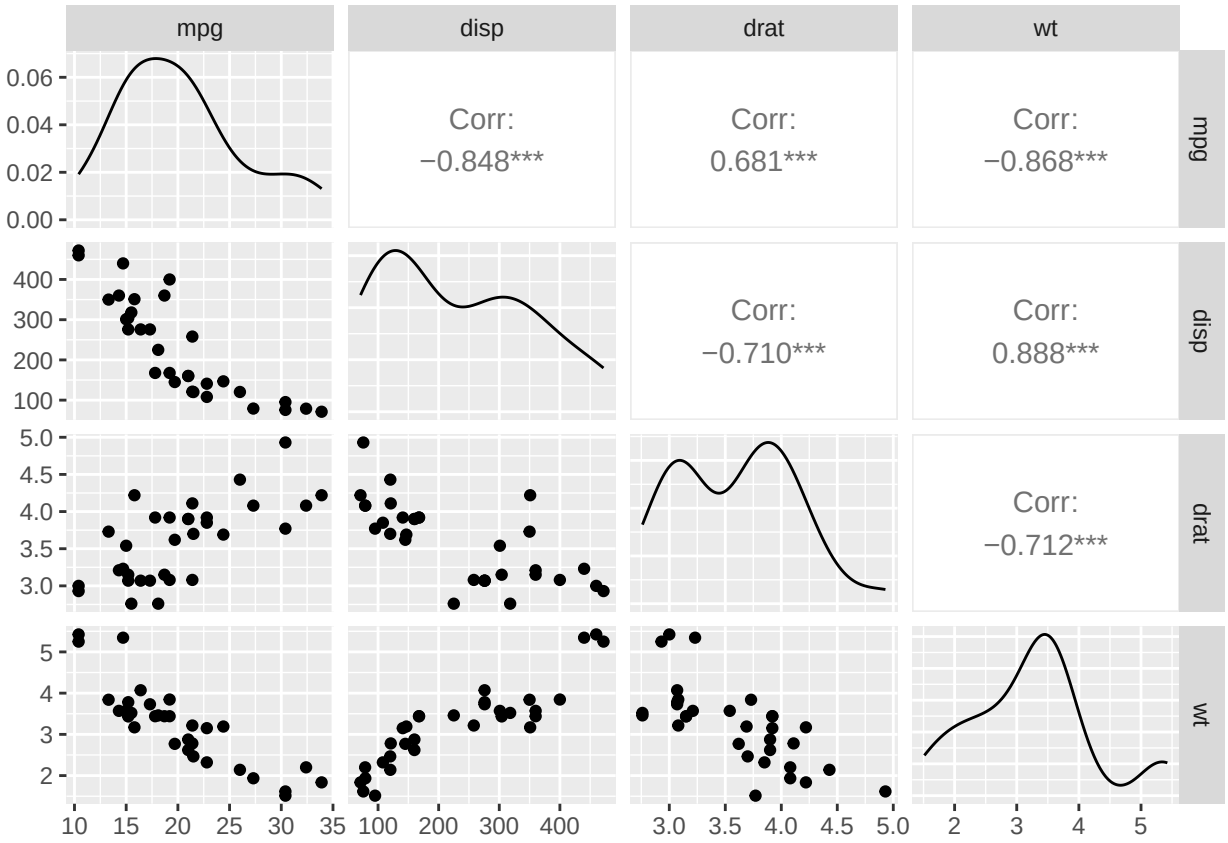
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



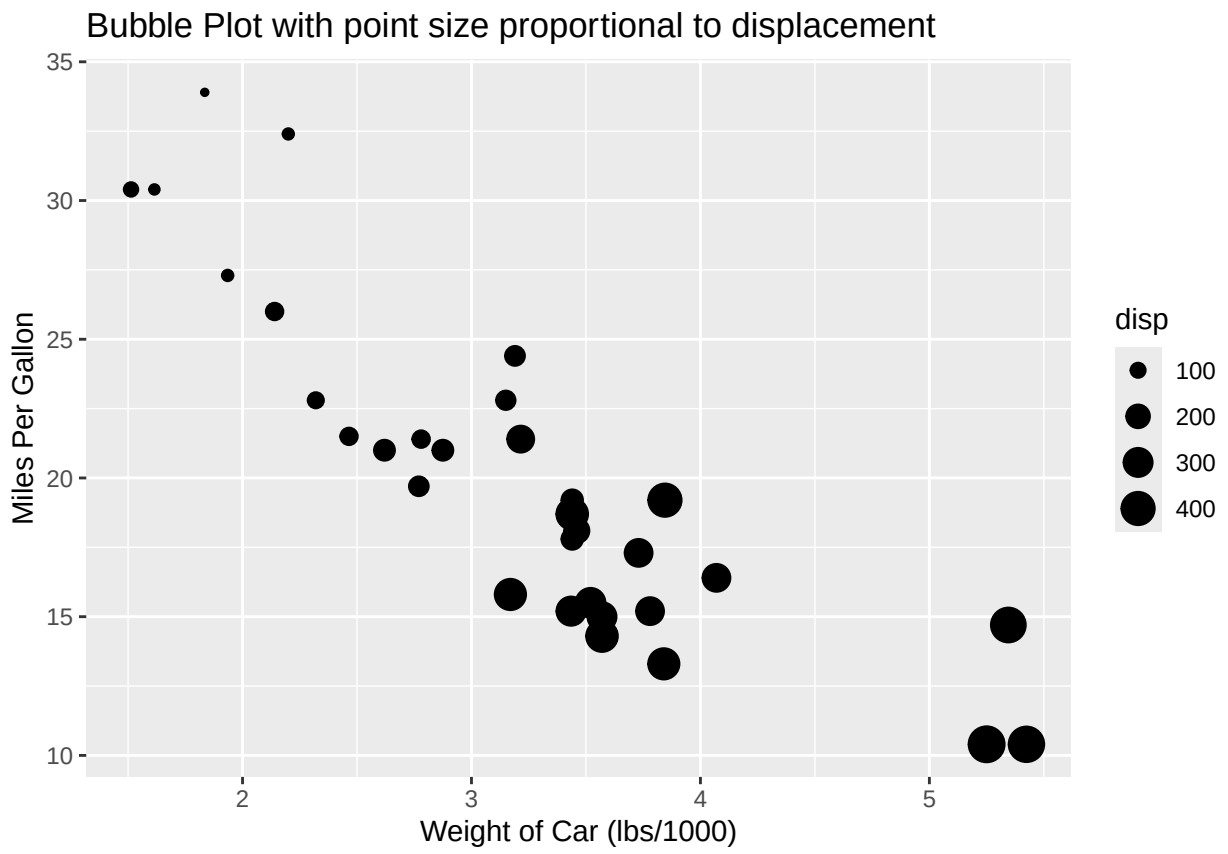

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

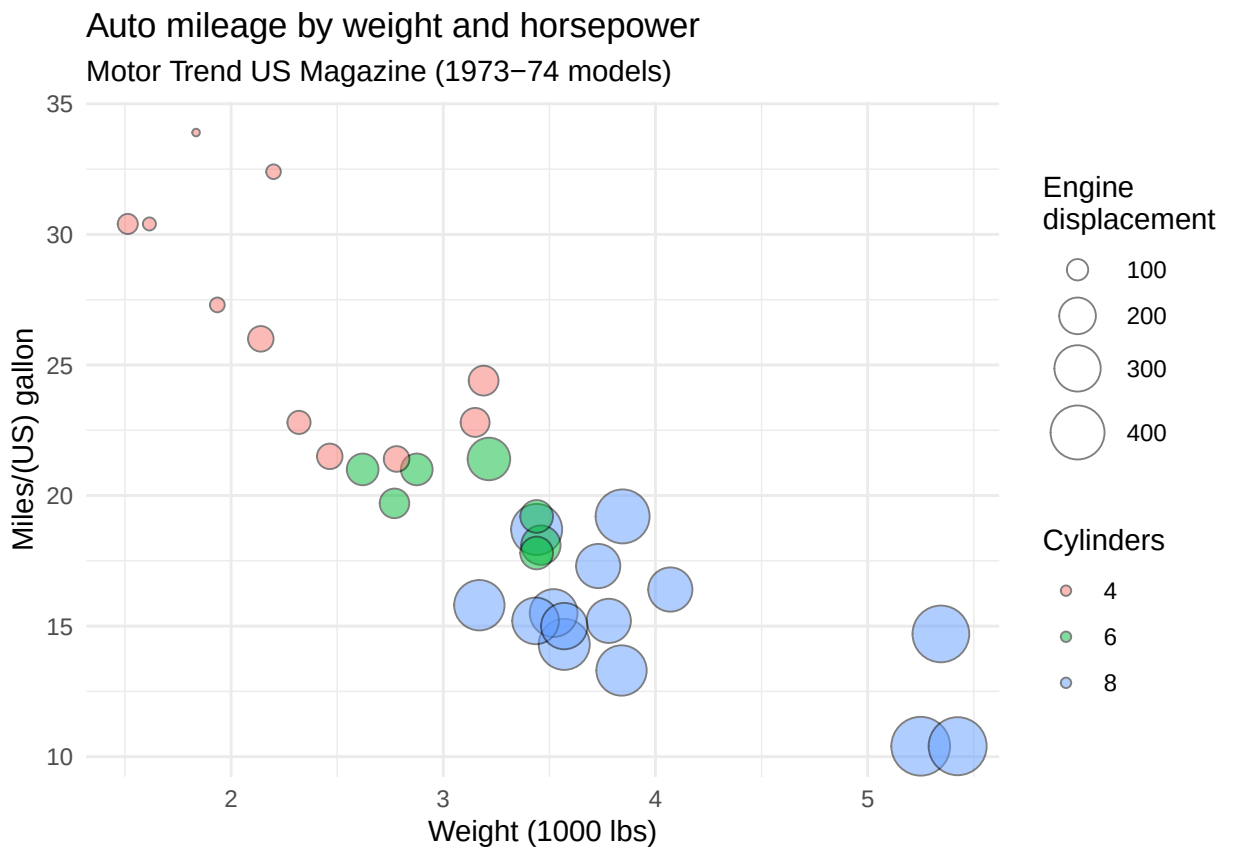
```
ggpairs(mtcars[c("mpg","disp","drat", "wt")])
```




```
mtcars |> ggplot(aes(x = wt, y = mpg, size = disp)) +
 geom_point() +
 labs(title="Bubble Plot with point size proportional to displacement",
 x="Weight of Car (lbs/1000)", y="Miles Per Gallon")
```




```
mtcars |> ggplot(aes(x = wt, y = mpg, size = disp,
 fill=factor(cyl))) +
 geom_point(alpha = .5, color = "black", shape = 21) +
 scale_size_continuous(range = c(1, 10)) +
 labs(title = "Auto mileage by weight and horsepower",
 subtitle = "Motor Trend US Magazine (1973-74 models)",
 x = "Weight (1000 lbs)",
 y = "Miles/(US) gallon", size = "Engine\ndisplacement",
 fill = "Cylinders") +
 theme_minimal()
```




```
library(corrgram)
```

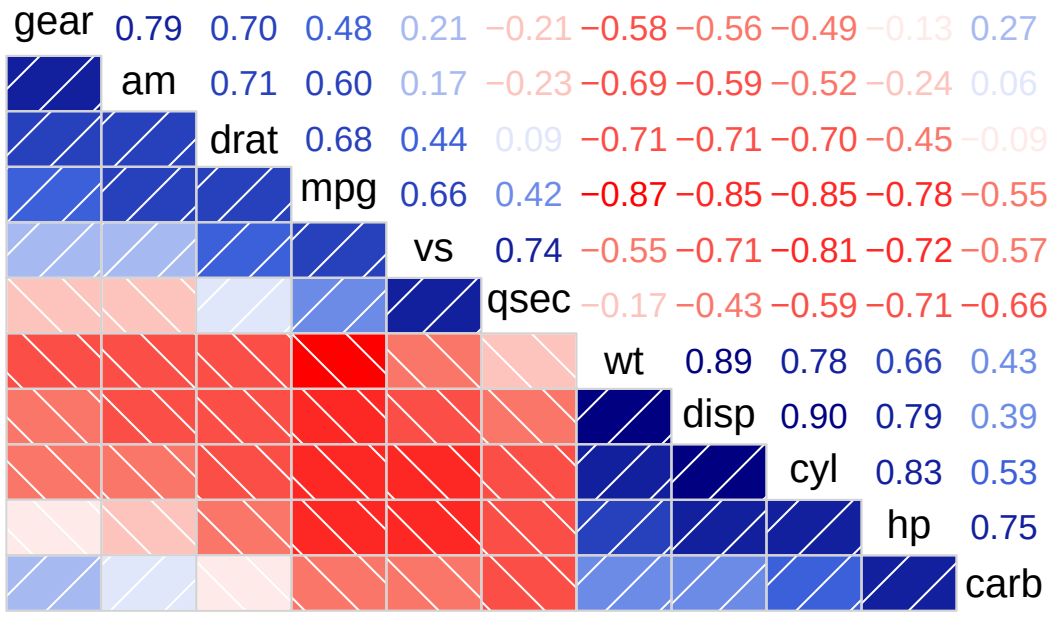
```
##
## Attaching package: 'corrgram'

## The following object is masked from 'package:GGally':
##
##     baseball
```

```
corrgram(mtcars, order=TRUE, lower.panel=panel.shade,
 upper.panel=panel.cor,
 main="Corrgram of mtcars data using shading and coefficients")
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```




```
# 列出所有合法参数
names(par())
```

```
##  [1] "xlog"      "ylog"      "adj"       "ann"       "ask"       "bg"
##  [7] "bty"       "cex"       "cex.axis"  "cex.lab"   "cex.main"  "cex.sub"
## [13] "cin"       "col"       "col.axis"  "col.lab"   "col.main"  "col.sub"
## [19] "cra"       "crt"       "csi"       "cxy"       "din"       "err"
## [25] "family"    "fg"        "fig"       "fin"       "font"      "font.axis"
```

```
## [31] "font.lab"  "font.main" "font.sub"  "lab"       "las"       "lend"
## [37] "lheight"   "ljoin"     "lmitre"    "lty"       "lwd"       "mai"
## [43] "mar"       "mex"       "mfcol"     "mfg"       "mfrow"     "mgp"
## [49] "mkh"       "new"       "oma"       "omd"       "omi"       "page"
## [55] "pch"       "pin"       "plt"       "ps"        "pty"       "smo"
## [61] "srt"       "tck"       "tcl"       "usr"       "xaxp"      "xaxs"
## [67] "xaxt"      "xpd"       "yaxp"      "yaxs"      "yaxt"      "ylbias"
```

```
# 检查 'usr' 是否可写
"usr" %in% names(par())    # TRUE（存在）
```

```
## [1] TRUE
```

```
"usr" %in% names(par(no.readonly = TRUE))  # FALSE（只读）
```

```
## [1] TRUE
```