ggplot2

```r
# load package(s) first
library(dplyr)
```

```
##
## 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
```

R has a sample dataframe “mtcars”.

```r
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9          27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2      26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa       30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora      15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E         21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

Using ggplot2 package

> ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details. (https://ggplot2.tidyverse.org)

RStudio ggplot2 Cheat Sheet

```r
# install.packages("ggplot2") # install only once
library(ggplot2) # load every session
```

Basic Syntax

`ggplot(data = dataset, mapping = aes(x = xcol, y = ycol)) + geom_histogram()`

- **ggplot layer**: create a ggplot object. especially `aes()` specifies what columns of the data table will be used as visual attributes of graphical elements in the plot.
- **geom layer**: define a shape of geometric plot
- and more other layers

Aesthetics

`colour`	Coloring outline
`fill`	Coloring inside
`linetype`	Line type
`shape`	Shape of point
`alpha`	Transparency

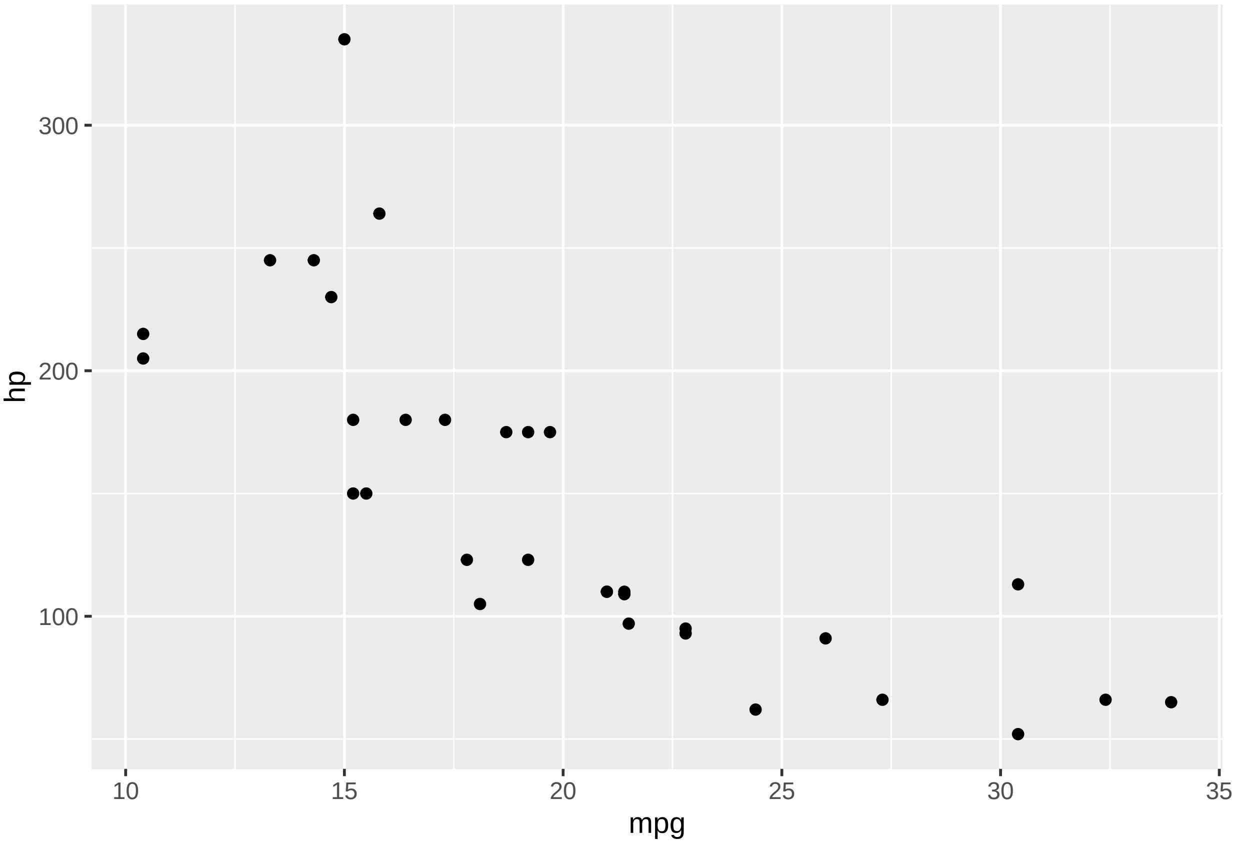
geom objects

`geom_point()`	Scatter plot
`geom_bar()`	Bar chart
`geom_line()`	Line plot
`geom_histogram()`	Histogram
`geom_boxplot()`	Box plot

Inside aes(): variables from dataframe.
Outside aes(): options not from dataframe. —

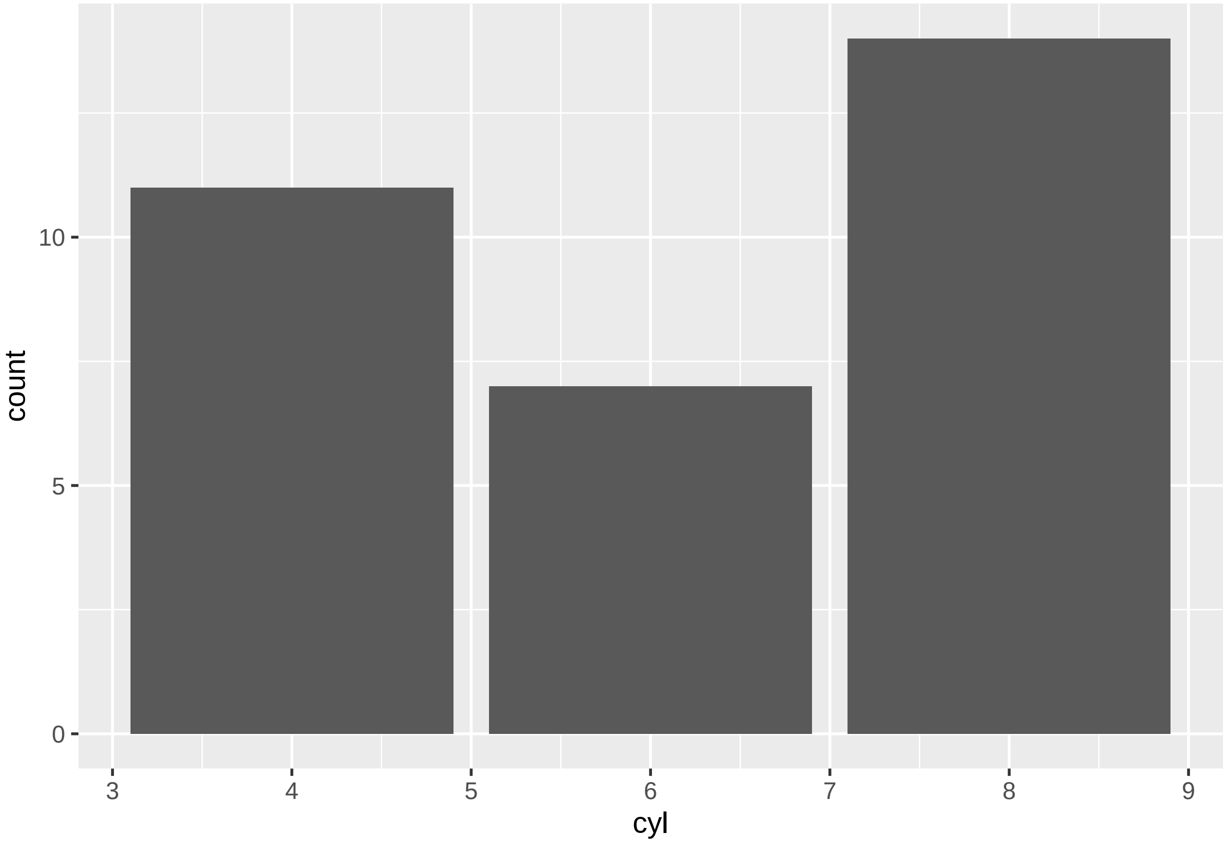
Scatter plot geom_point()

```
ggplot(mtcars, aes(x = mpg, y = hp)) + geom_point()
```



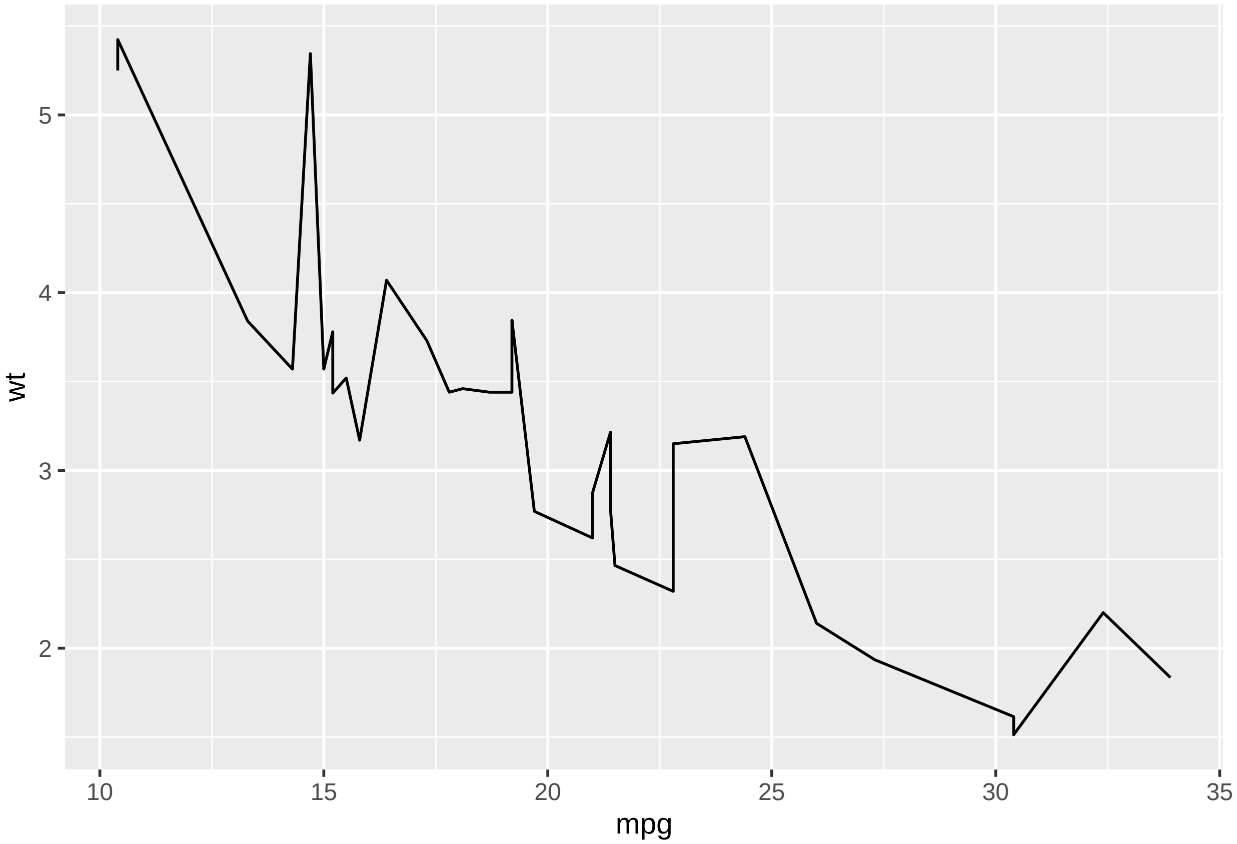
Bar chart geom_bar()

```
ggplot(mtcars, aes(x = cyl)) + geom_bar()
```



Line plot `geom_line()`

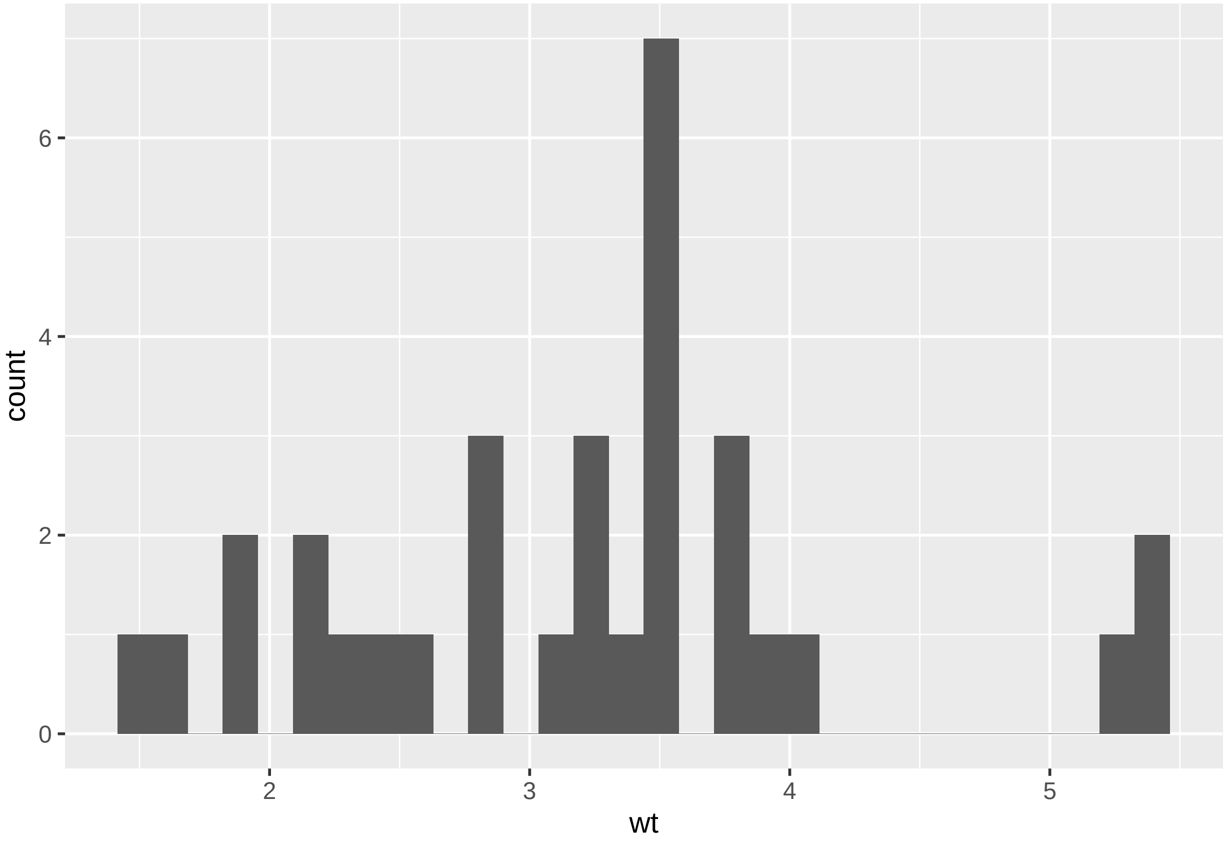
```r
ggplot(mtcars, aes(x = mpg, y = wt)) + geom_line()
```



Histogram `geom_histogram()`

```
ggplot(mtcars, aes(x = wt)) + geom_histogram()
```

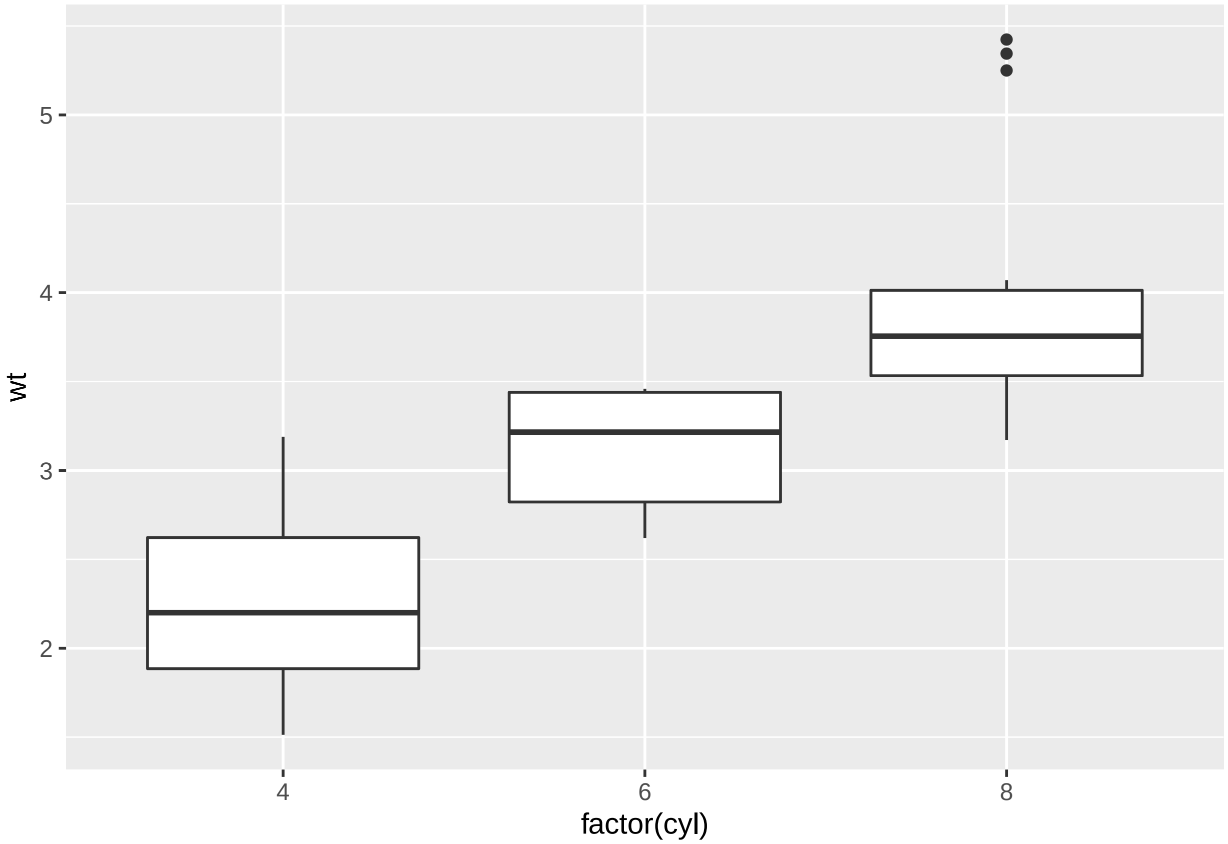
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



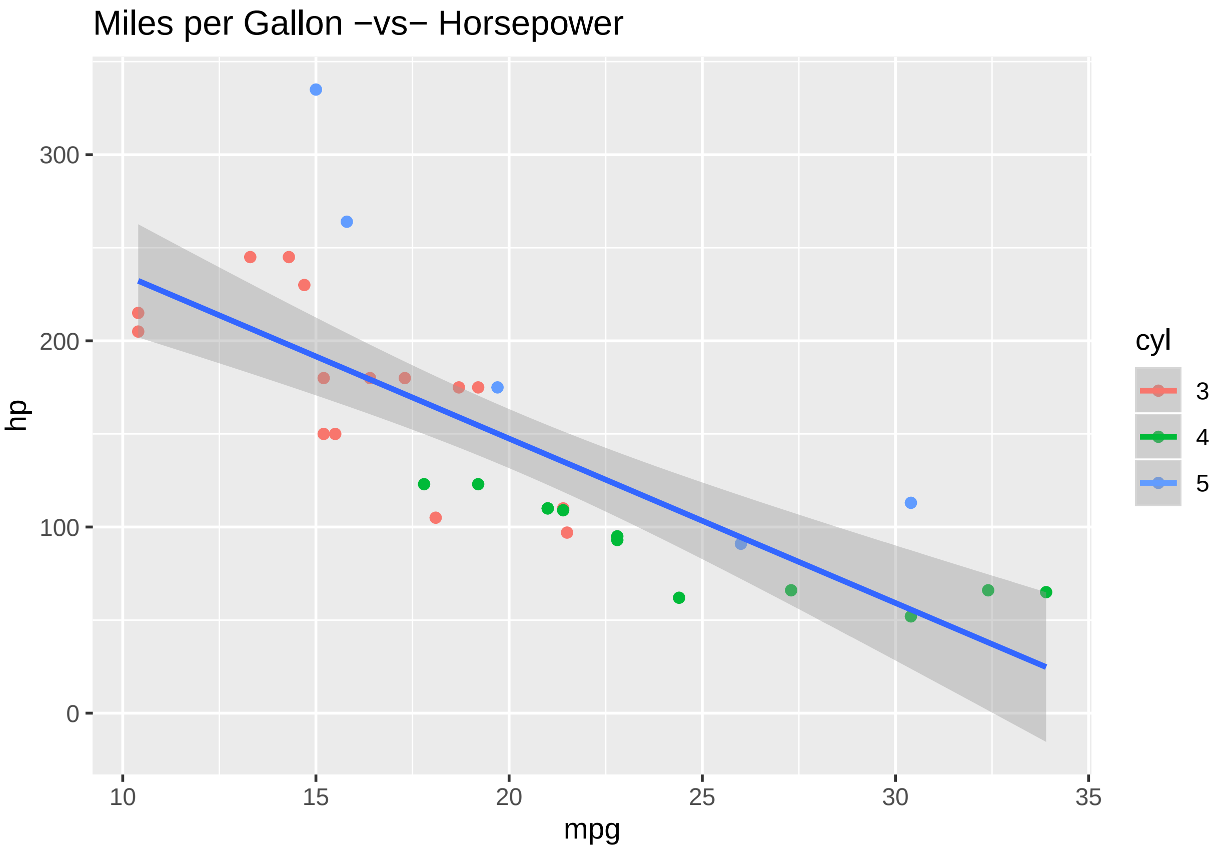
Box plot `geom_boxplot()`

Use `factor()` to treat `cyl` as a discrete (categorical) variable.

```
ggplot(mtcars, aes(x = factor(cyl), y = wt)) + geom_boxplot()
```



```
ggplot(mtcars, aes(x = mpg, y = hp, colour = cyl)) +
  geom_point(aes(color = factor(gear))) +
  geom_smooth(method = "lm") +
  labs(title = "Miles per Gallon -vs- Horsepower")
```



---

storms dataframe

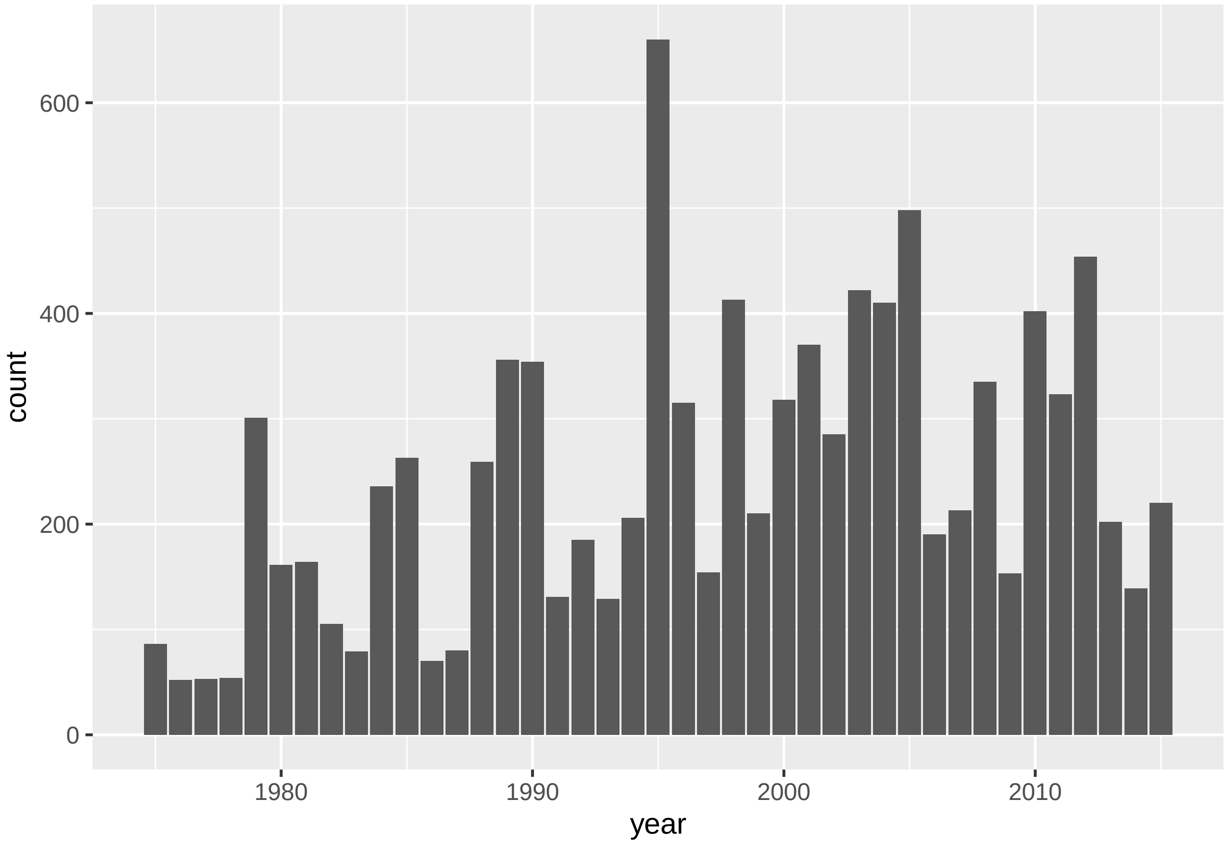
```
head(storms)
```

```
## # A tibble: 6 x 13
##   name   year month   day  hour   lat  long status category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>  <ord>    <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropi~ -1          25     1013
## 2 Amy    1975     6    27     6  28.5 -79   tropi~ -1          25     1013
## 3 Amy    1975     6    27    12  29.5 -79   tropi~ -1          25     1013
## 4 Amy    1975     6    27    18  30.5 -79   tropi~ -1          25     1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropi~ -1          25     1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropi~ -1          25     1012
## # ... with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

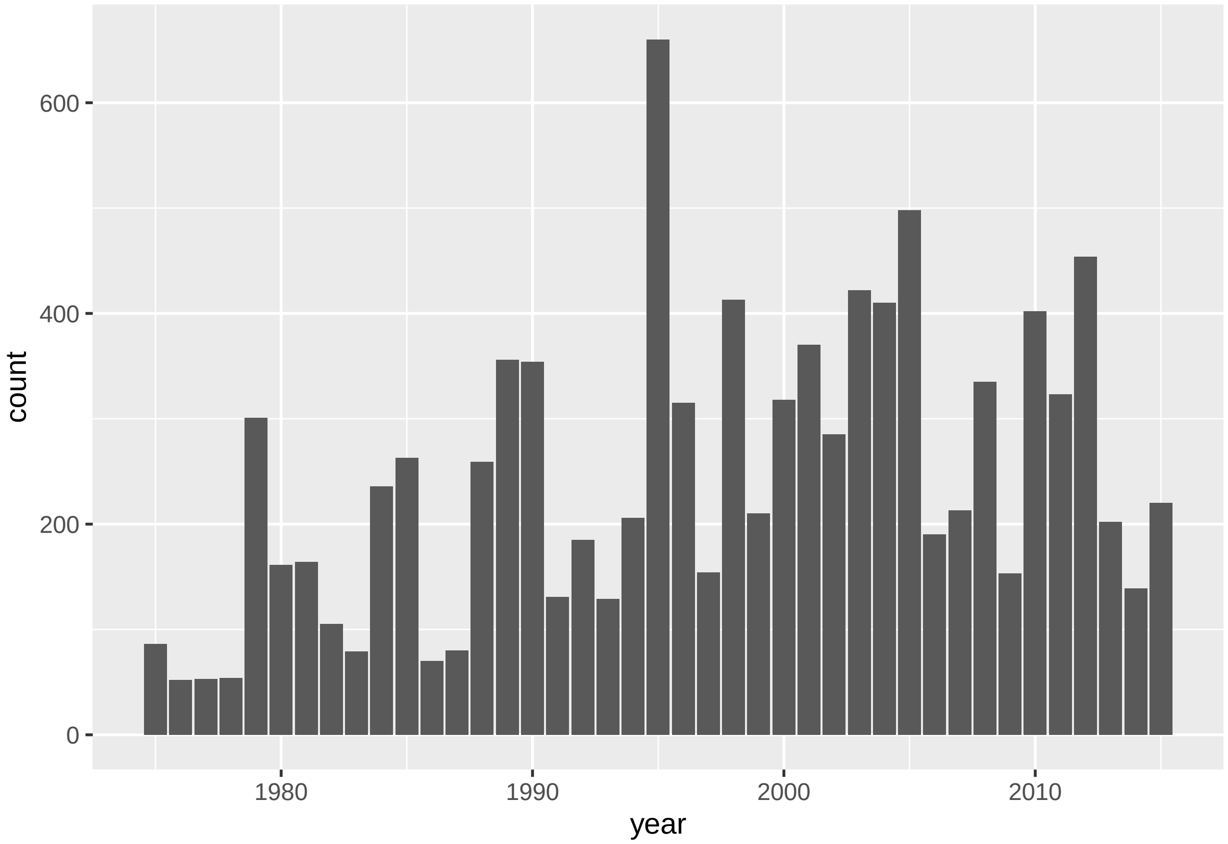
Bar plot

How many records are there in each year?

```
ggplot(storms, aes(x = year)) + geom_bar()
```



```r
# this works as well
ggplot(storms) + geom_bar(aes(x = year))
```



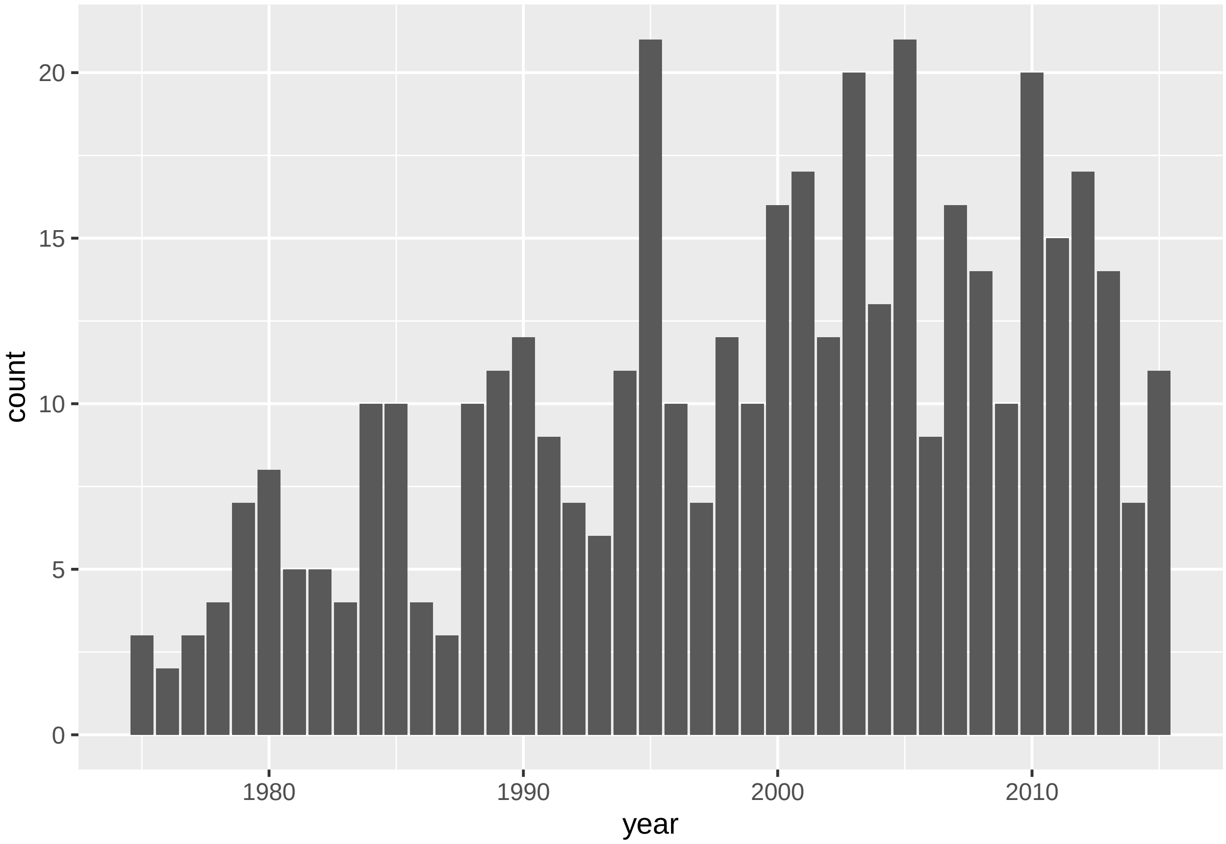
Then, how many storms are there in each year?

Need some operation.

```
distinct(group_by(select(storms, year, name), year))
```

```
## # A tibble: 426 x 2
## # Groups:   year [41]
##     year name
##    <dbl> <chr>
##  1  1975 Amy
##  2  1975 Caroline
##  3  1975 Doris
##  4  1976 Belle
##  5  1976 Gloria
##  6  1977 Anita
##  7  1977 Clara
##  8  1977 Evelyn
##  9  1978 Amelia
## 10  1978 Bess
## # ... with 416 more rows
```

```
storms_year_name <- distinct(group_by(select(storms, year, name), year))

ggplot(storms_year_name) + geom_bar(aes(x = year))
```



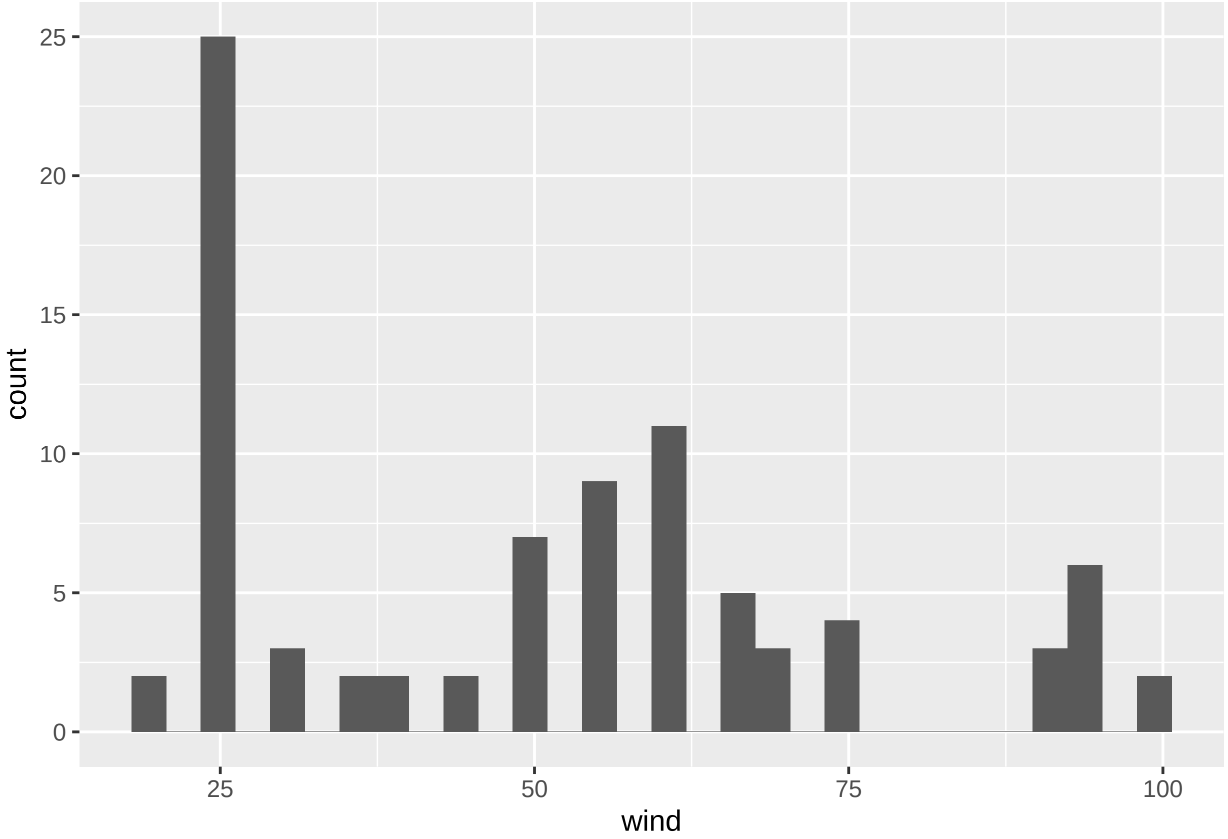
```
# check
count(storms_year_name)
```

```
## # A tibble: 41 x 2
## # Groups:   year [41]
##     year     n
##    <dbl> <int>
##  1  1975     3
##  2  1976     2
##  3  1977     3
##  4  1978     4
##  5  1979     7
##  6  1980     8
##  7  1981     5
##  8  1982     5
##  9  1983     4
## 10  1984    10
## # ... with 31 more rows
```

Histogram

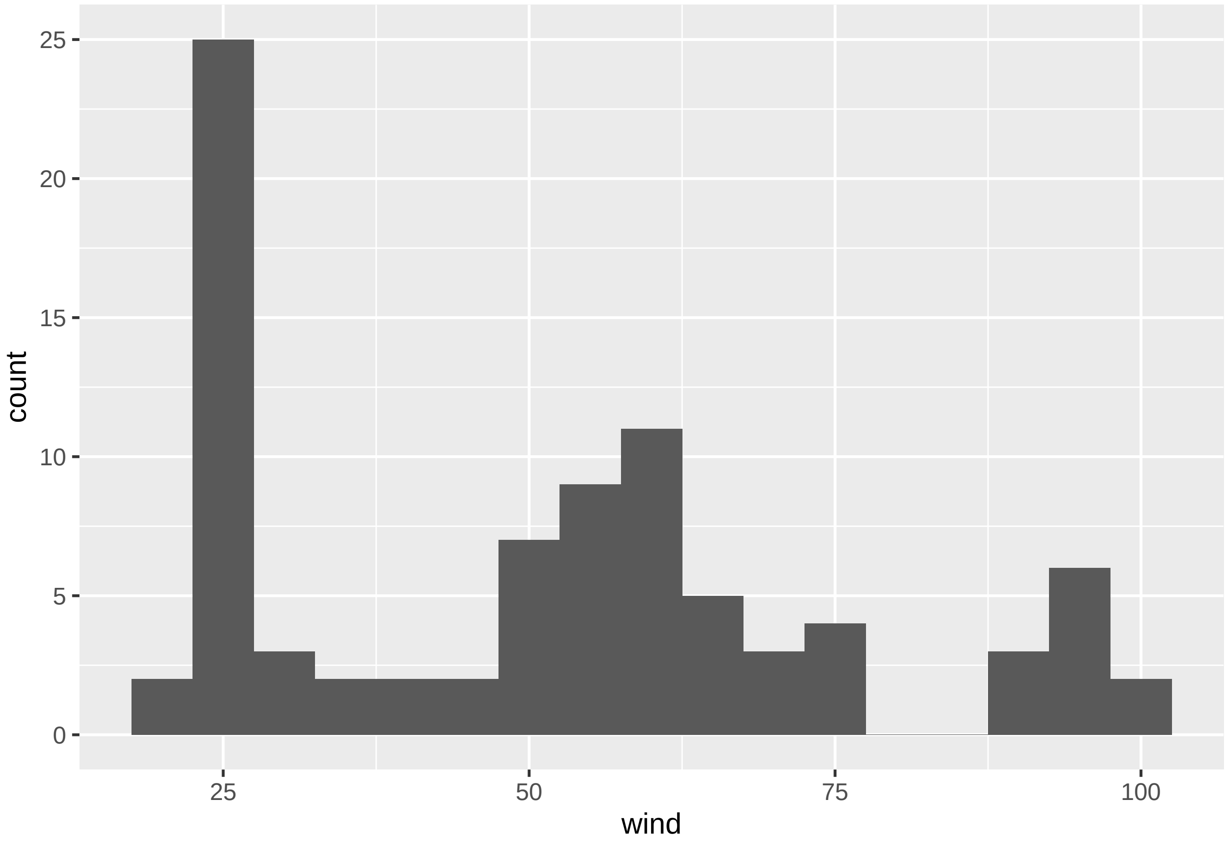
```
storms75 <- filter(storms, year == 1975)

ggplot(storms75) + geom_histogram(aes(x = wind))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

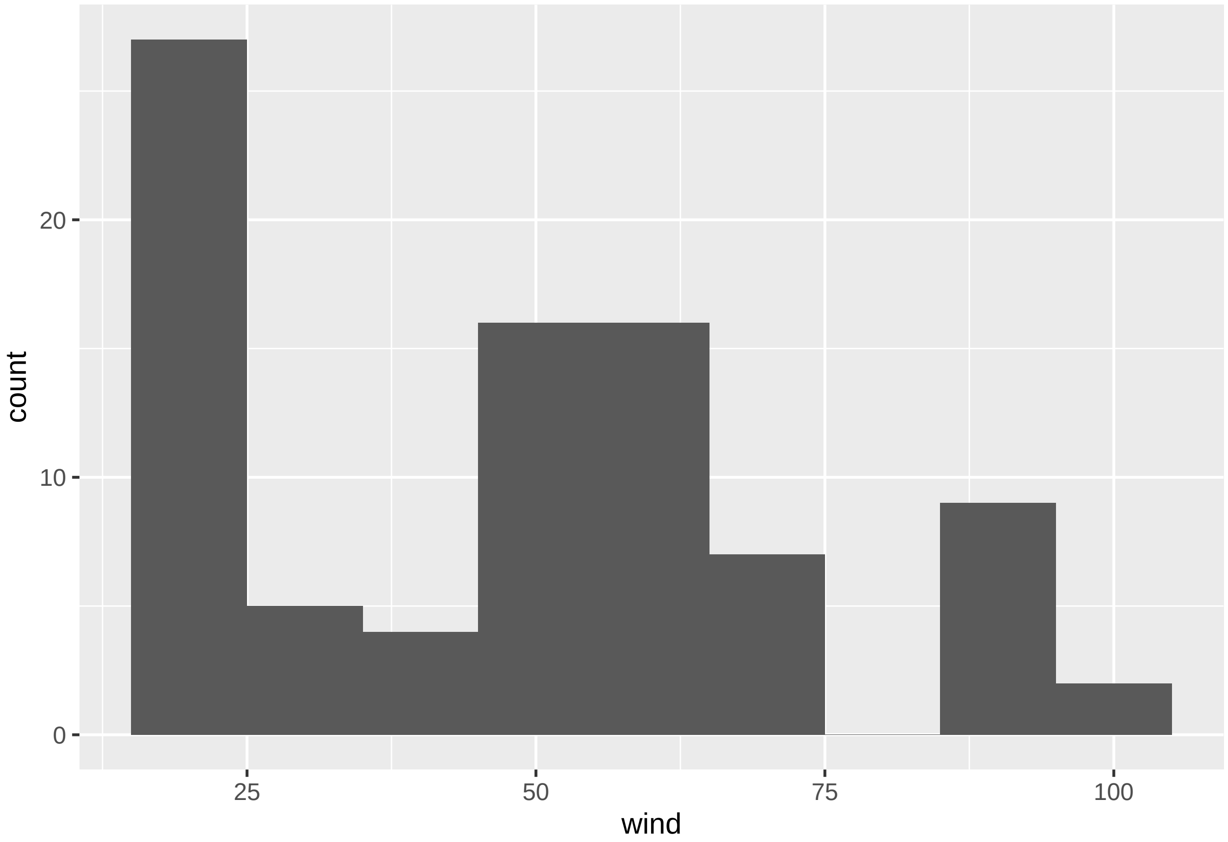


Change the bin width and compare.

```r
ggplot(storms75, aes(x = wind)) + geom_histogram(binwidth = 5)
```



```
ggplot(storms75, aes(x = wind)) + geom_histogram(binwidth = 10)
```



Box plot

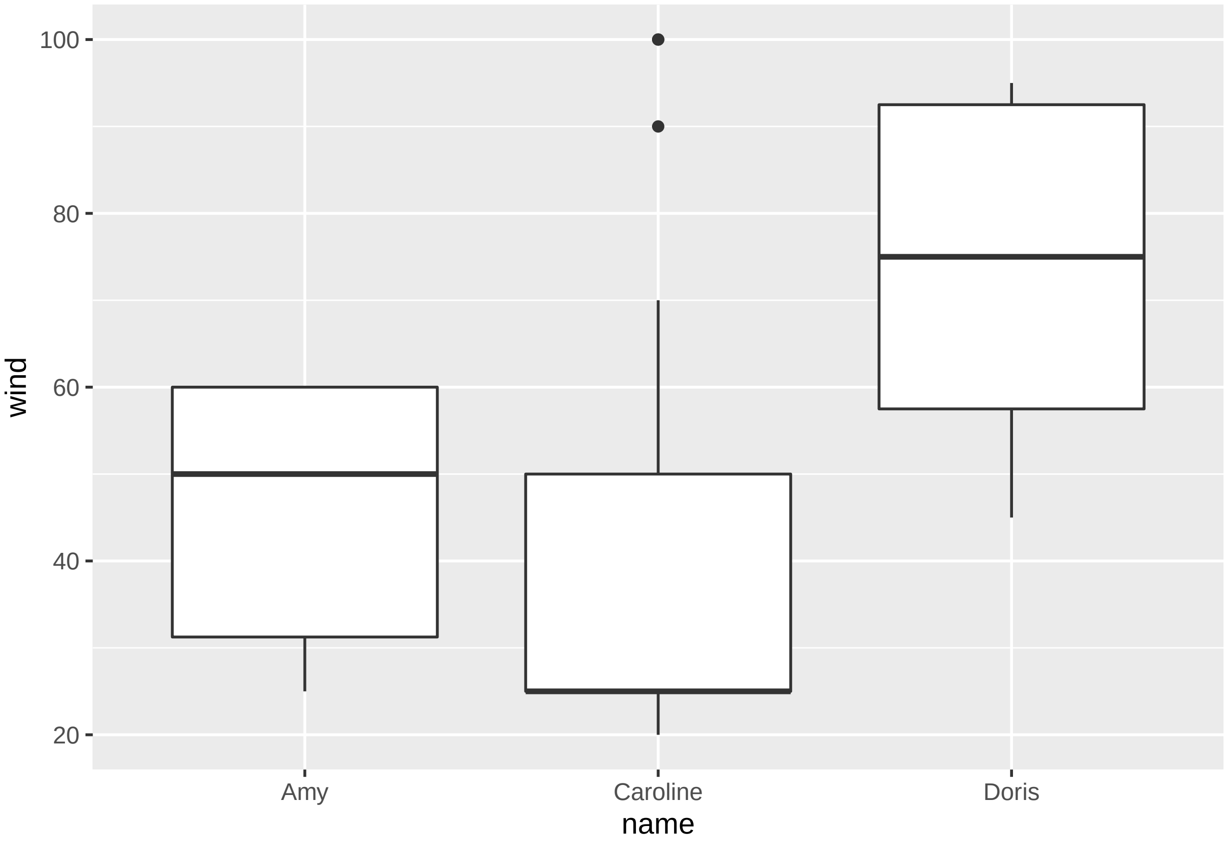
There are three storms in 1975: Amy, Caroline, and Doris.

```
unique(pull(storms75, name))
```

```
## [1] "Amy"      "Caroline" "Doris"
```

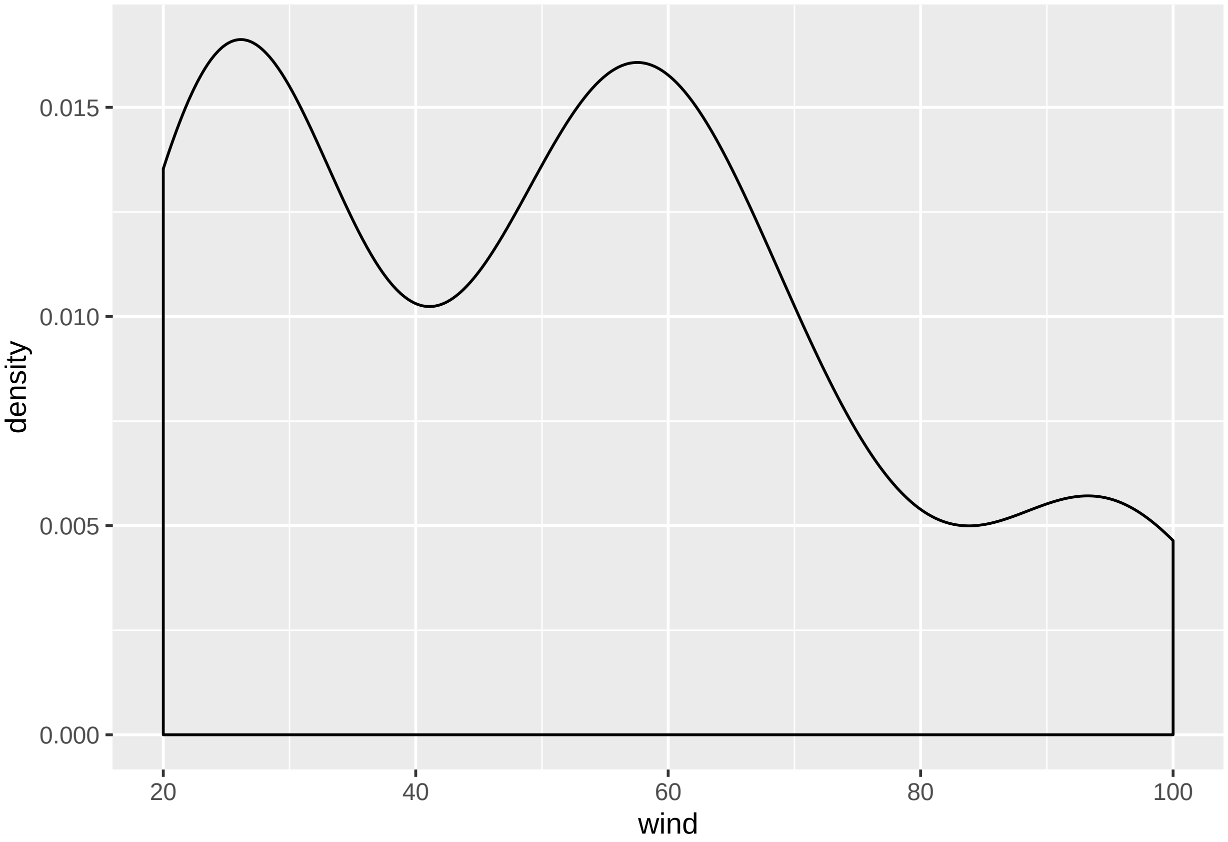
Compare the wind speeds of the three.

```
ggplot(storms75, aes(x = name, y = wind)) + geom_boxplot()
```



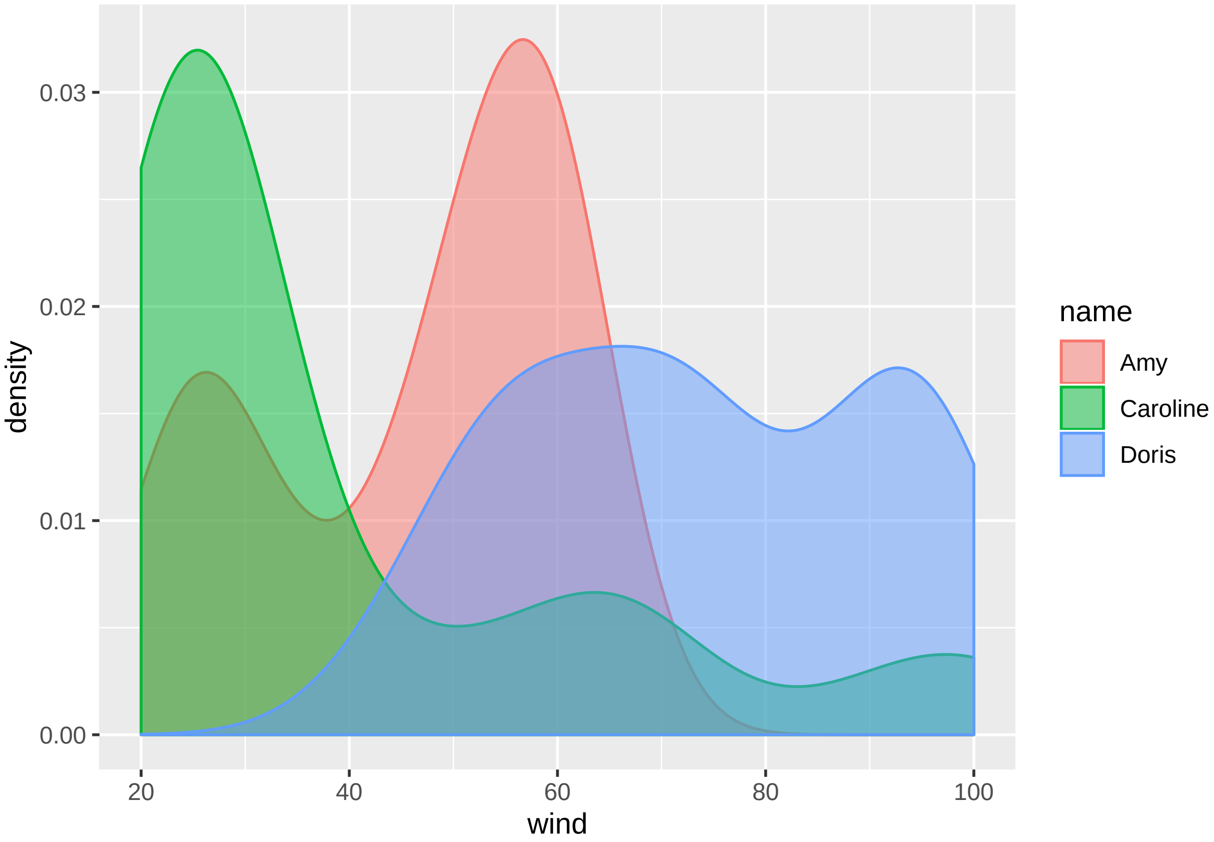
Density curve

```
ggplot(storms75, aes(x = wind)) + geom_density()
```



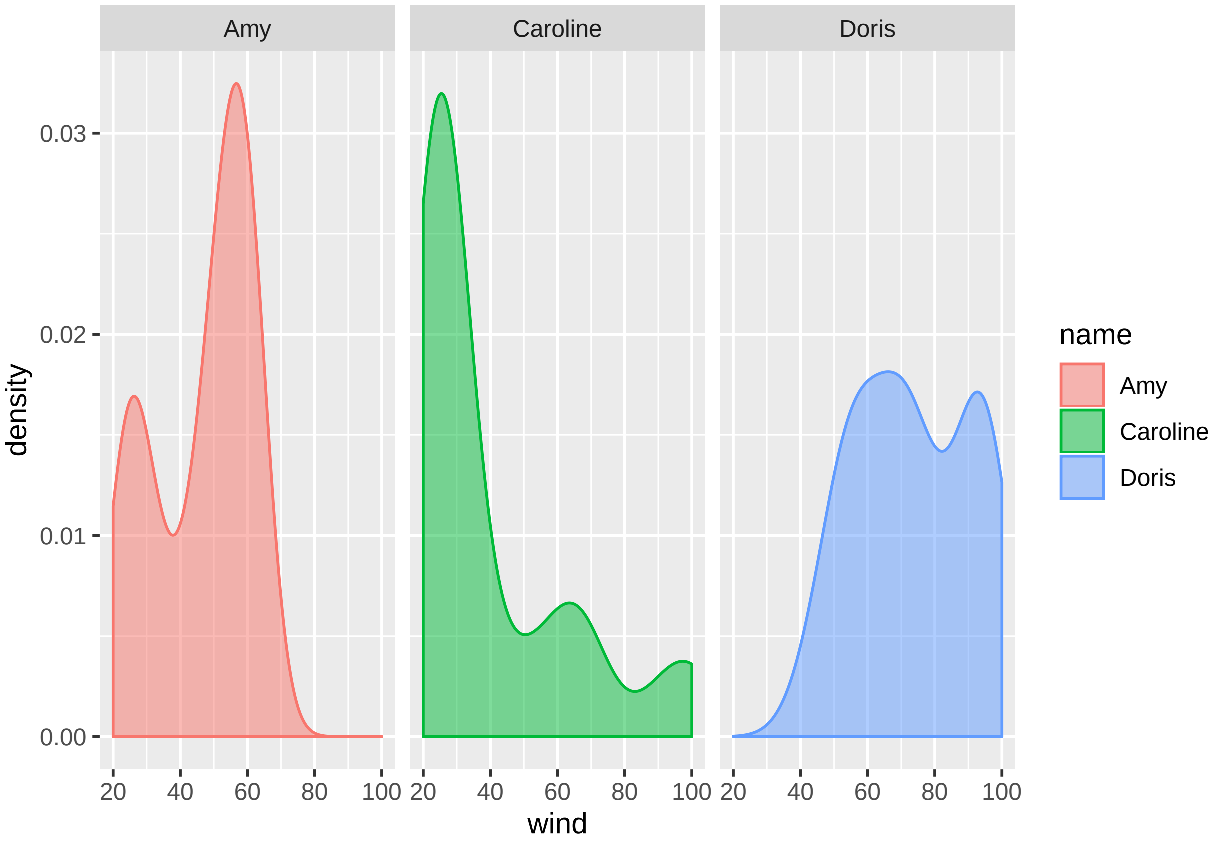
How is the distribution like?

```
ggplot(storms75, aes(x = wind, color = name)) +
  geom_density(aes(fill = name), alpha = 0.5)
```



To produce separated frames, use `facet_wrap()`. Facetting by `name`.

```
ggplot(storms75, aes(x = wind, color = name)) +
  geom_density(aes(fill = name), alpha = 0.5) +
  facet_wrap(~ name)
```

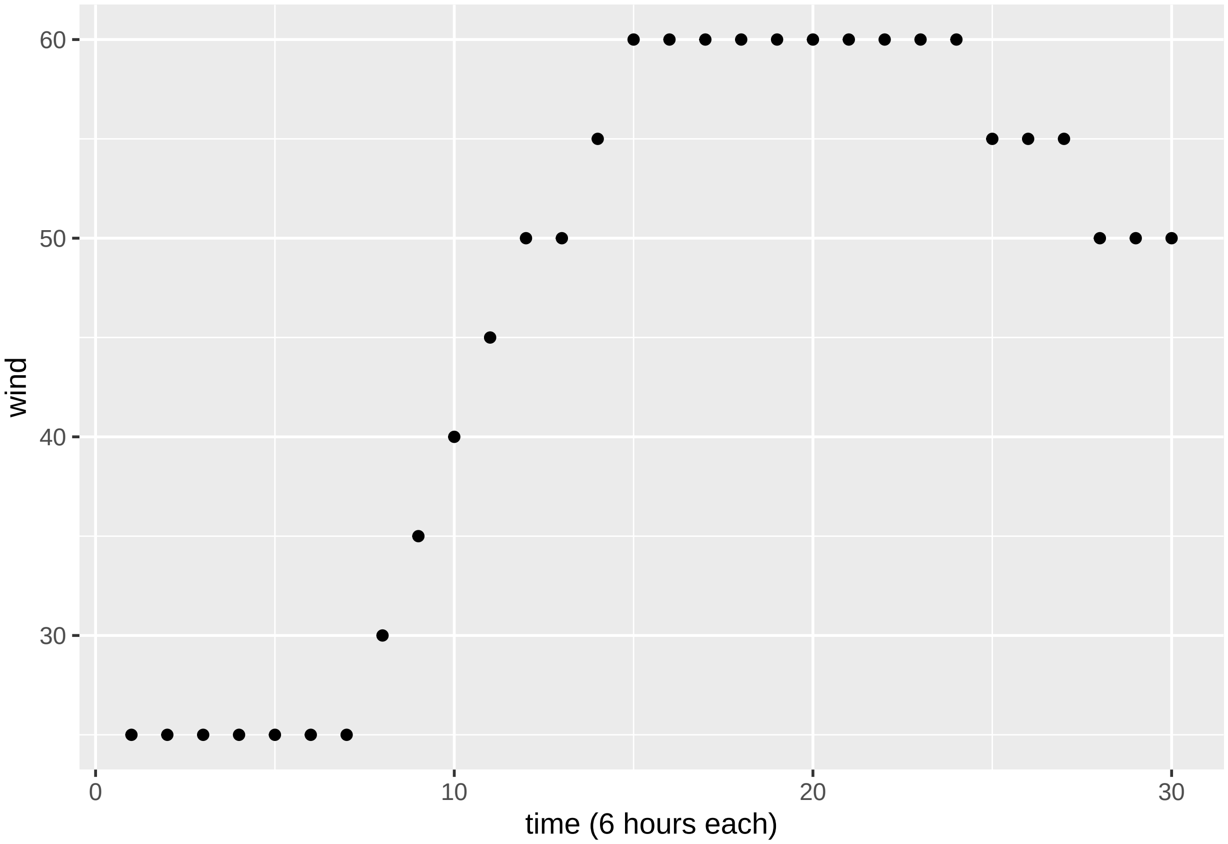


Scatter plot

```
amy75 <- filter(storms75, name == "Amy")
head(amy75)
```

```
## # A tibble: 6 x 13
##   name   year month   day  hour   lat  long status category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>  <ord>    <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropi~ -1          25     1013
## 2 Amy    1975     6    27     6  28.5 -79   tropi~ -1          25     1013
## 3 Amy    1975     6    27    12  29.5 -79   tropi~ -1          25     1013
## 4 Amy    1975     6    27    18  30.5 -79   tropi~ -1          25     1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropi~ -1          25     1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropi~ -1          25     1012
## # ... with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

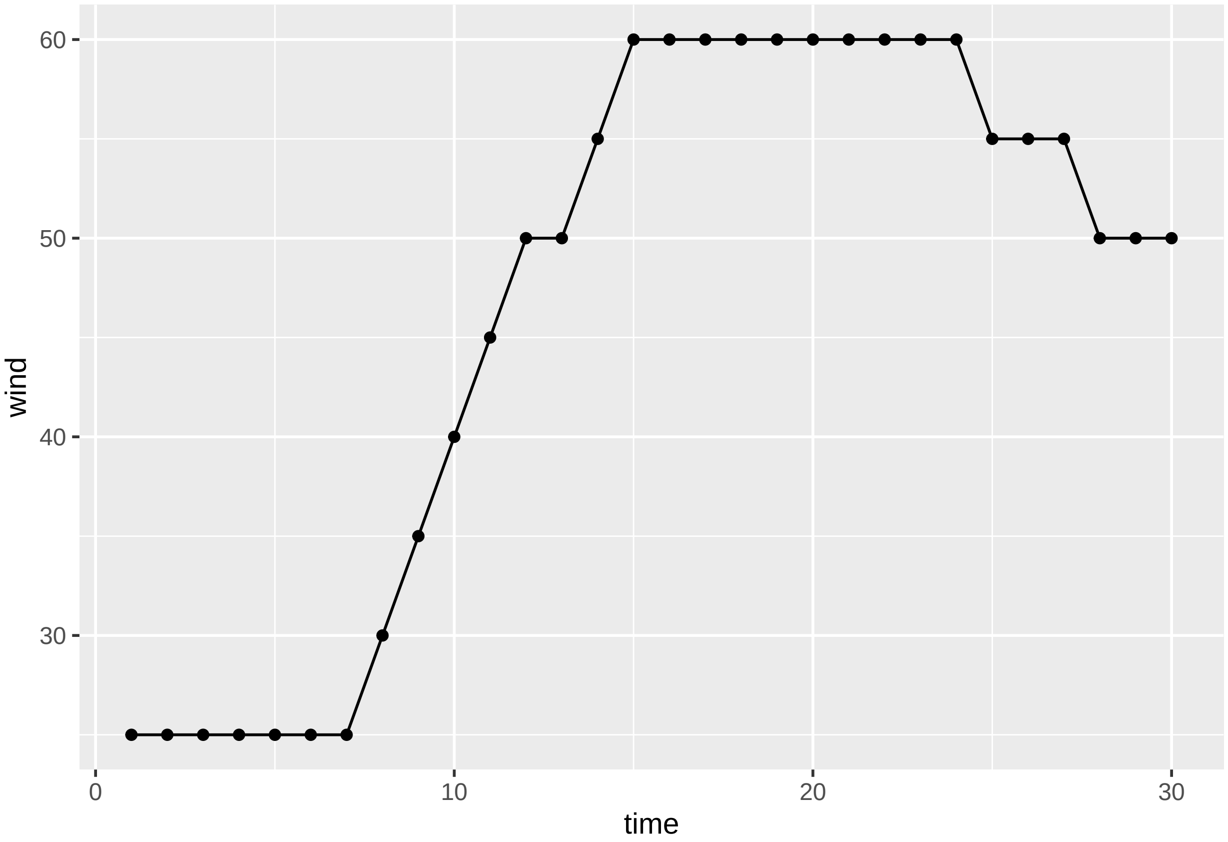
```
ggplot(data = amy75, aes(x = 1:nrow(amy75), y = wind)) +
  geom_point() +
  xlab("time (6 hours each)")
```



Line plot

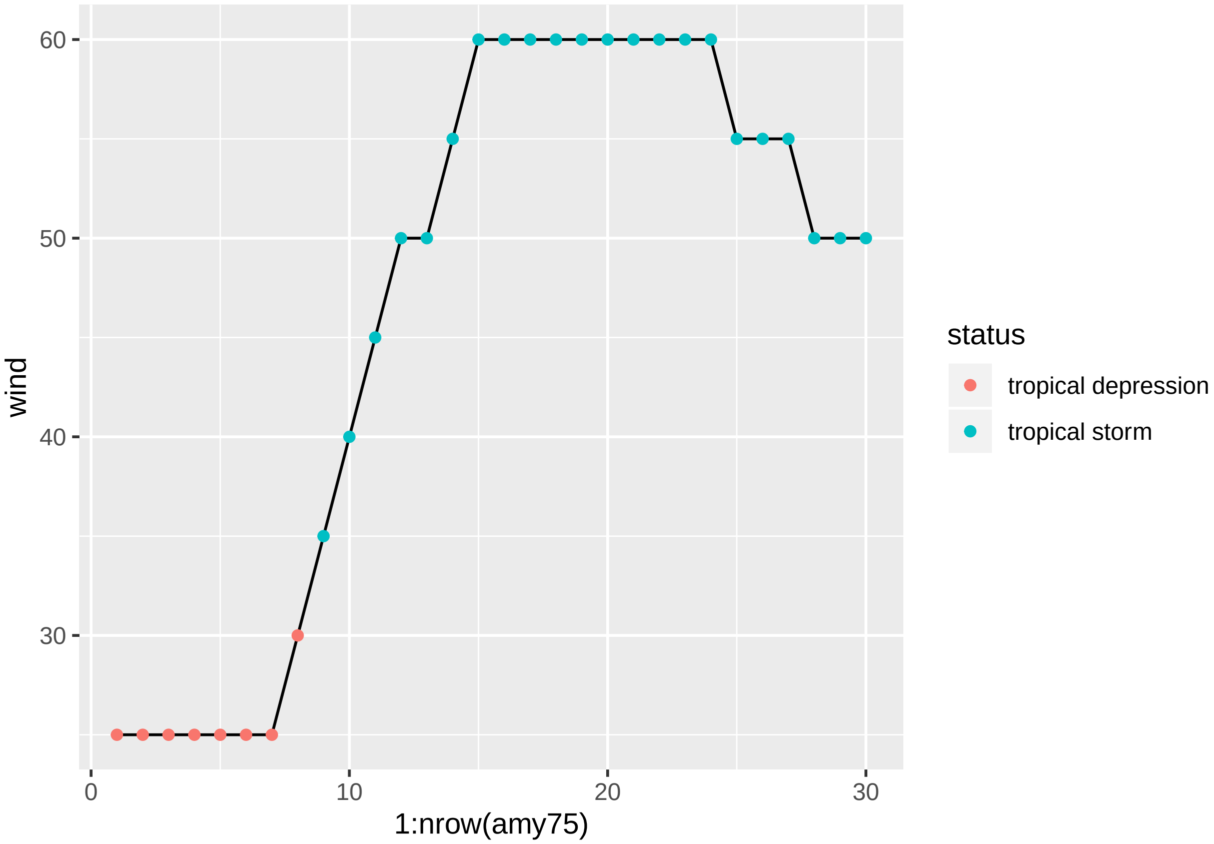
For chronological graph, line plot is commonly used.

```
ggplot(data = amy75, aes(x = 1:nrow(amy75), y = wind)) +
  geom_point() +
  geom_line() +
  xlab("time")
```



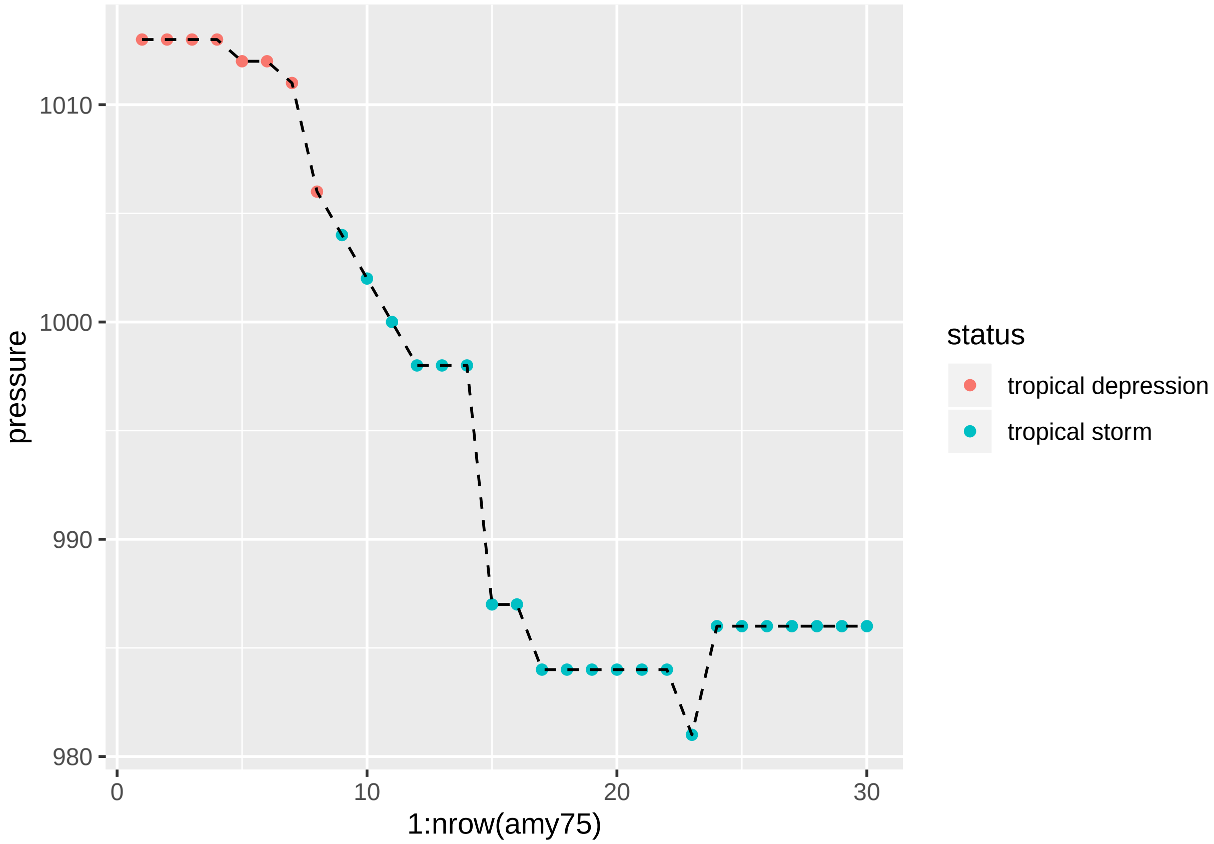
Color by `status`.

```
ggplot(amy75, aes(x = 1:nrow(amy75), y = wind)) +
  geom_line() +
  geom_point(aes(color = status))
```



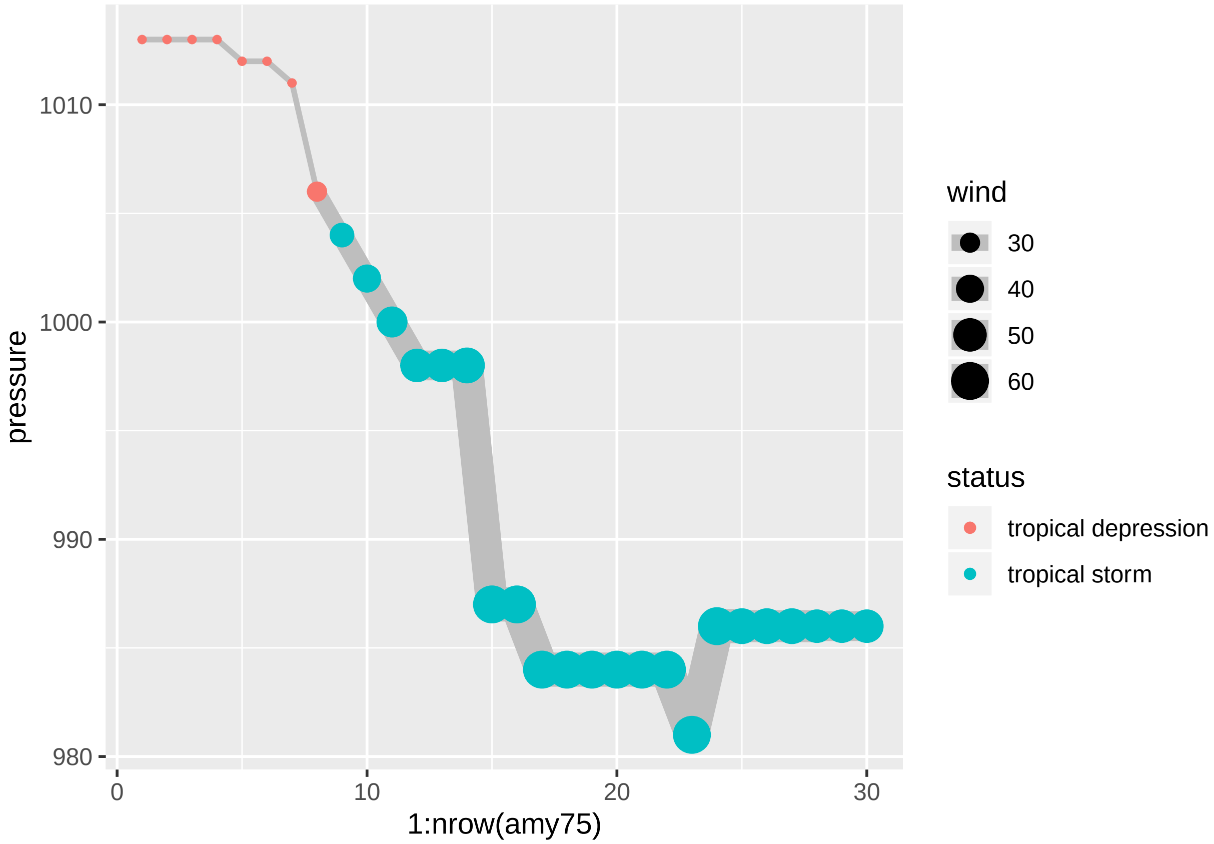
What about pressure?

```
ggplot(amy75, aes(x = 1:nrow(amy75), y = pressure)) +
  geom_point(aes(color = status)) +
  geom_line(linetype = "dashed")
```



Graphing pressure and taking into account the wind speed reflected in the size of points and line segments.

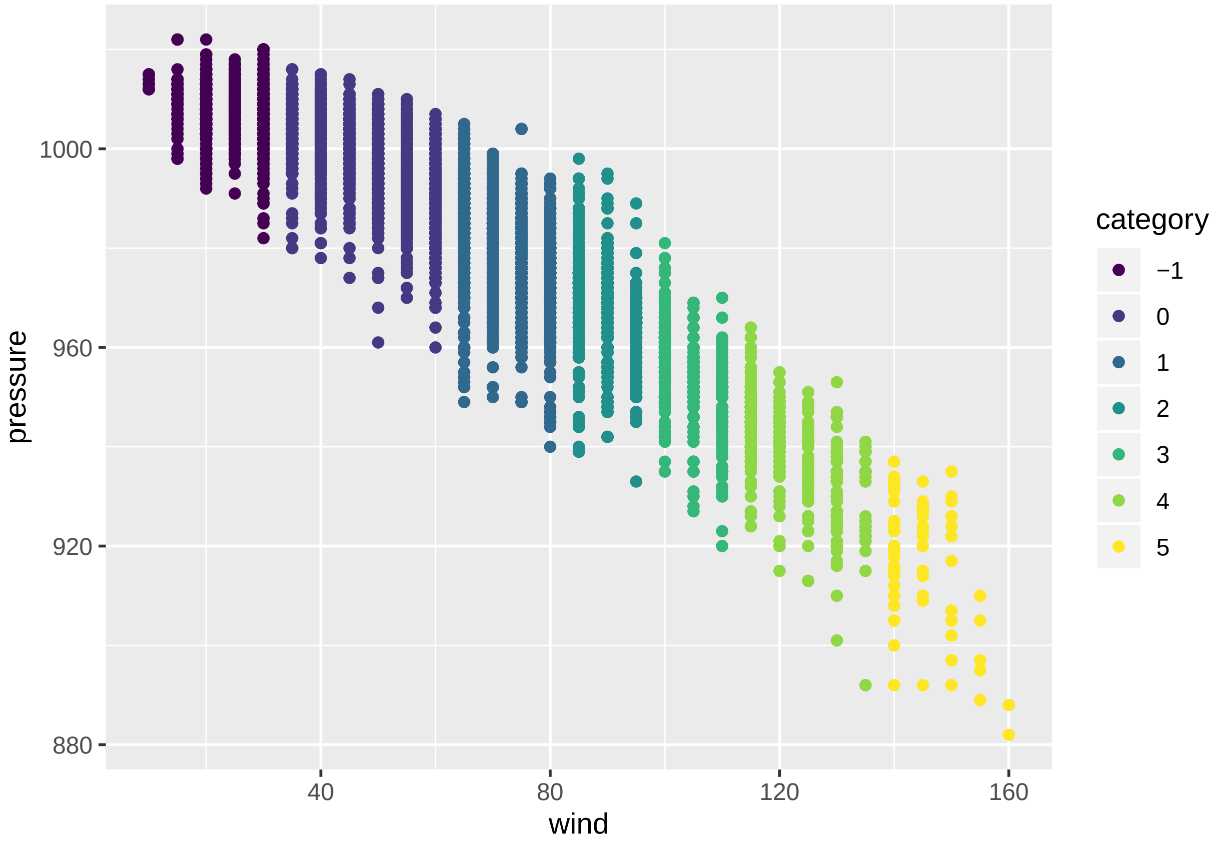
```r
ggplot(amy75, aes(x = 1:nrow(amy75), y = pressure)) +
  geom_line(aes(size = wind), color = "gray") +
  geom_point(aes(color = status, size = wind))
```



Exercise

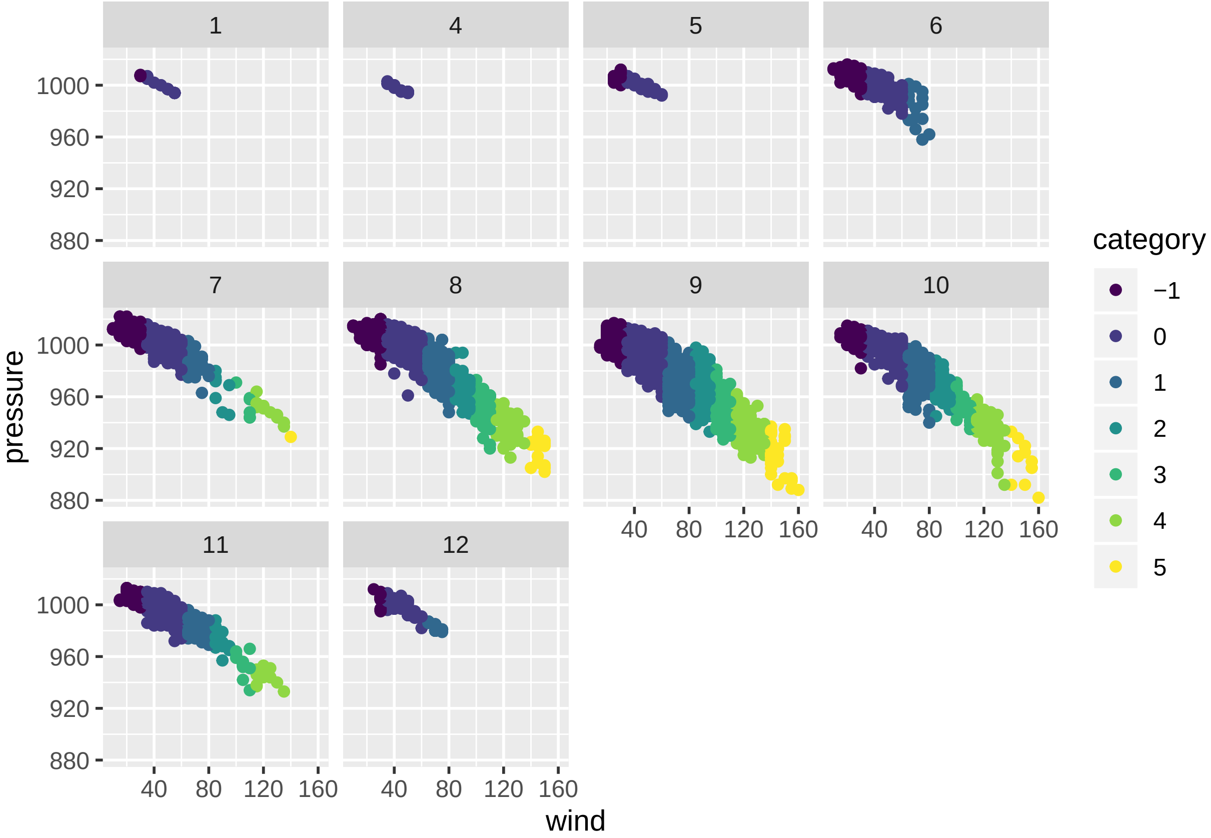
1) Use "ggplot2" functions to make a single scatterplot of wind and pressure for all storms. Use category to add color to the dots.

```
ggplot(storms, aes(x = wind, y = pressure)) +
  geom_point(aes(color = category))
```



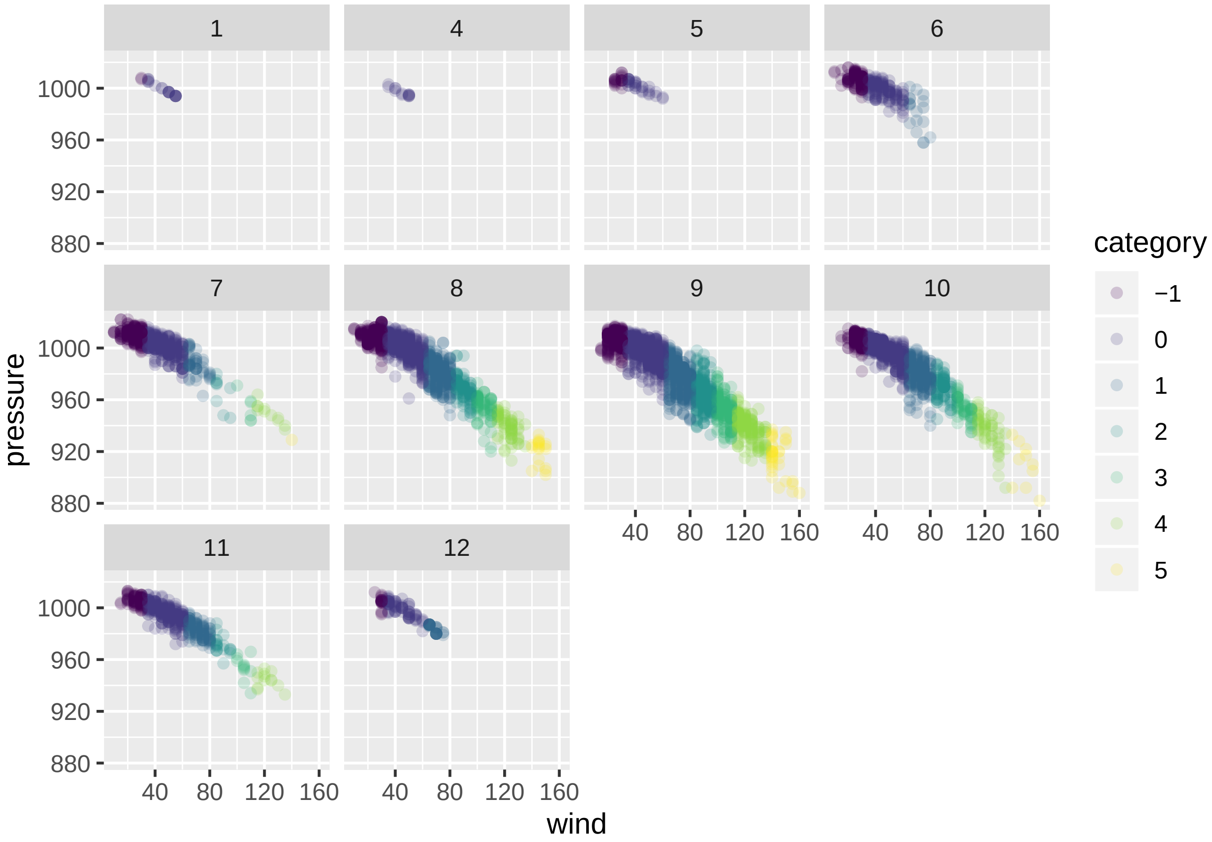
2) Use “ggplot2” functions to make a scatterplot of wind and pressure for all storms, facetting by month, and using category to differentiate by color.

```
ggplot(storms, aes(x = wind, y = pressure)) +
  geom_point(aes(color = category)) +
  facet_wrap(~ month)
```



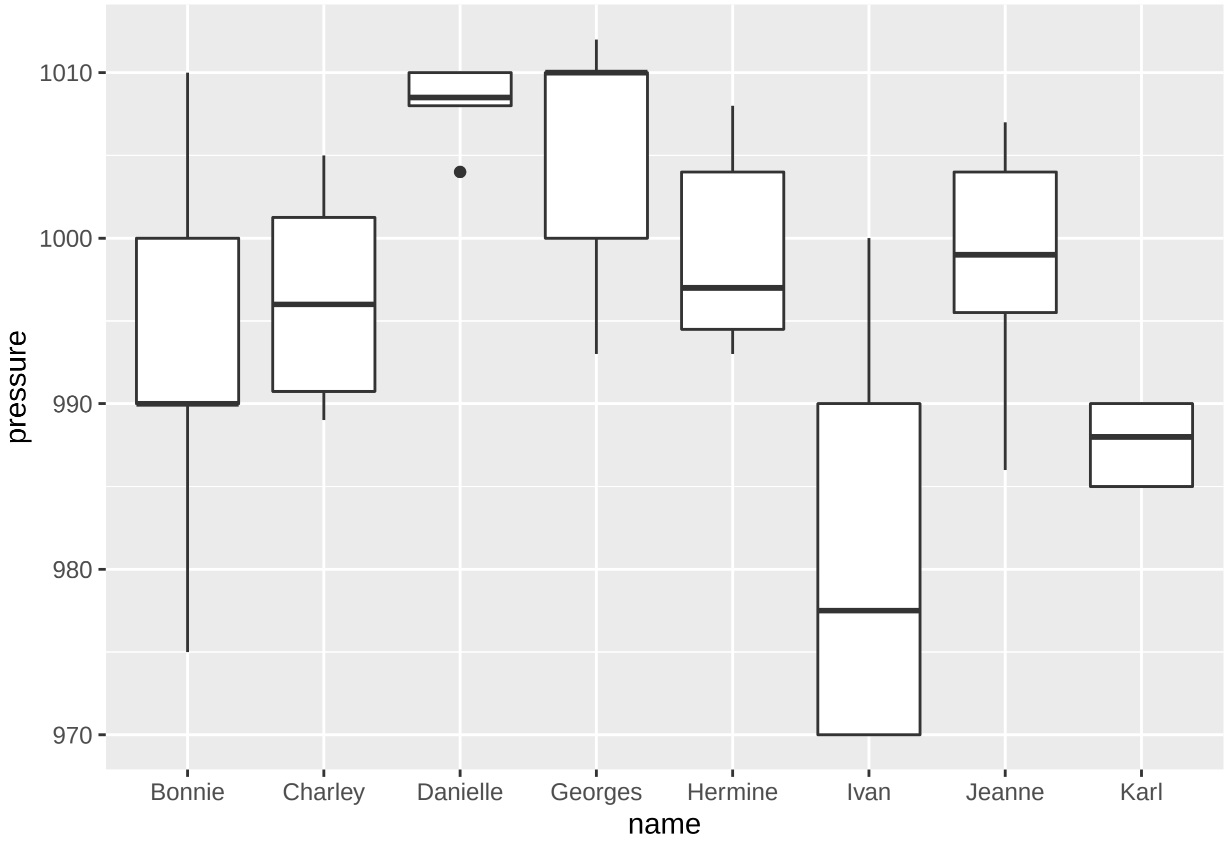
3) Use “ggplot2” functions to make a scatterplot of wind and pressure for all storms, but now create facets based on month. Feel free to add some amount of alpha transparency to the color of dots.

```
ggplot(storms, aes(x = wind, y = pressure)) +
  geom_point(aes(color = category), alpha = 0.2) +
  facet_wrap(~ month)
```

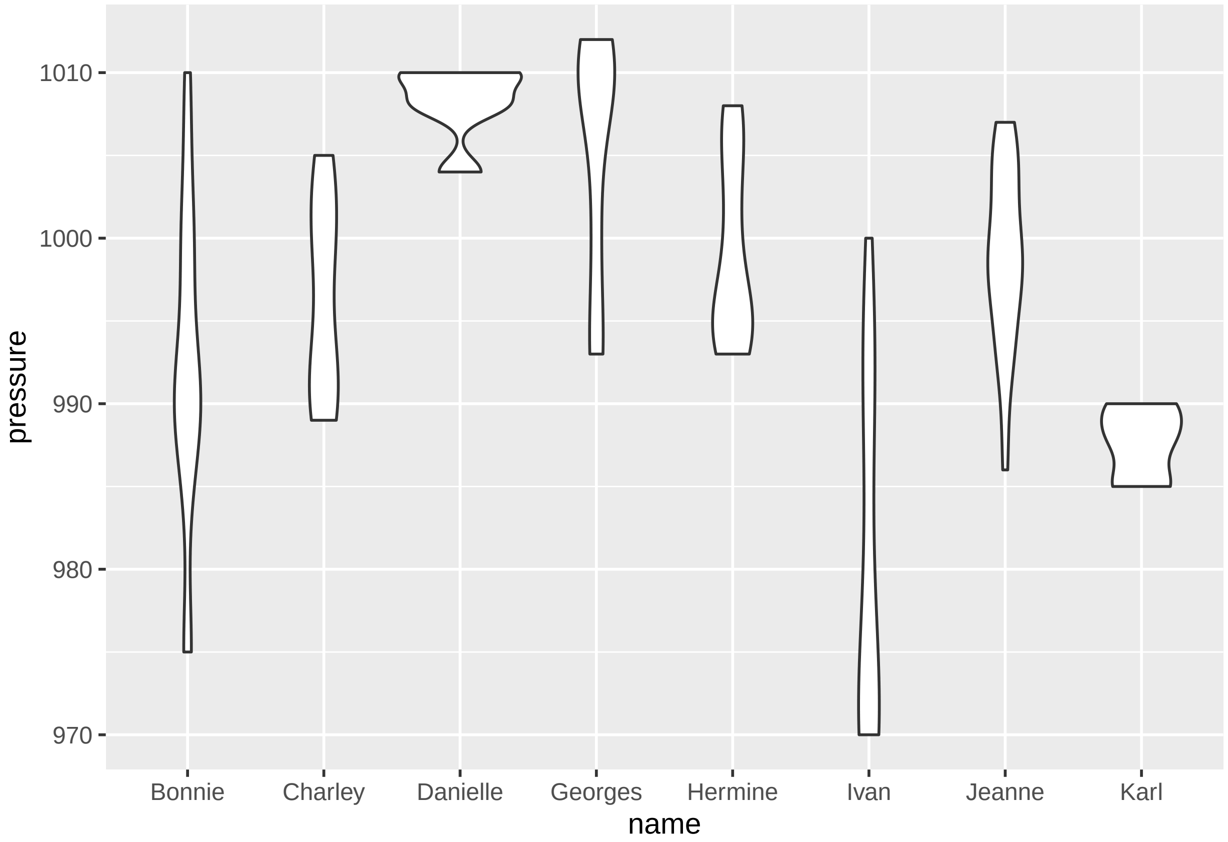


4) Create boxplots of pressure, for storms in 1980. You can also try graphing violins (geom_violin()) instead of boxplots (geom_boxplot()).

```r
storms80 <- filter(storms, year == 1980)

ggplot(storms80, aes(x = name, y = pressure)) +
  geom_boxplot()
```

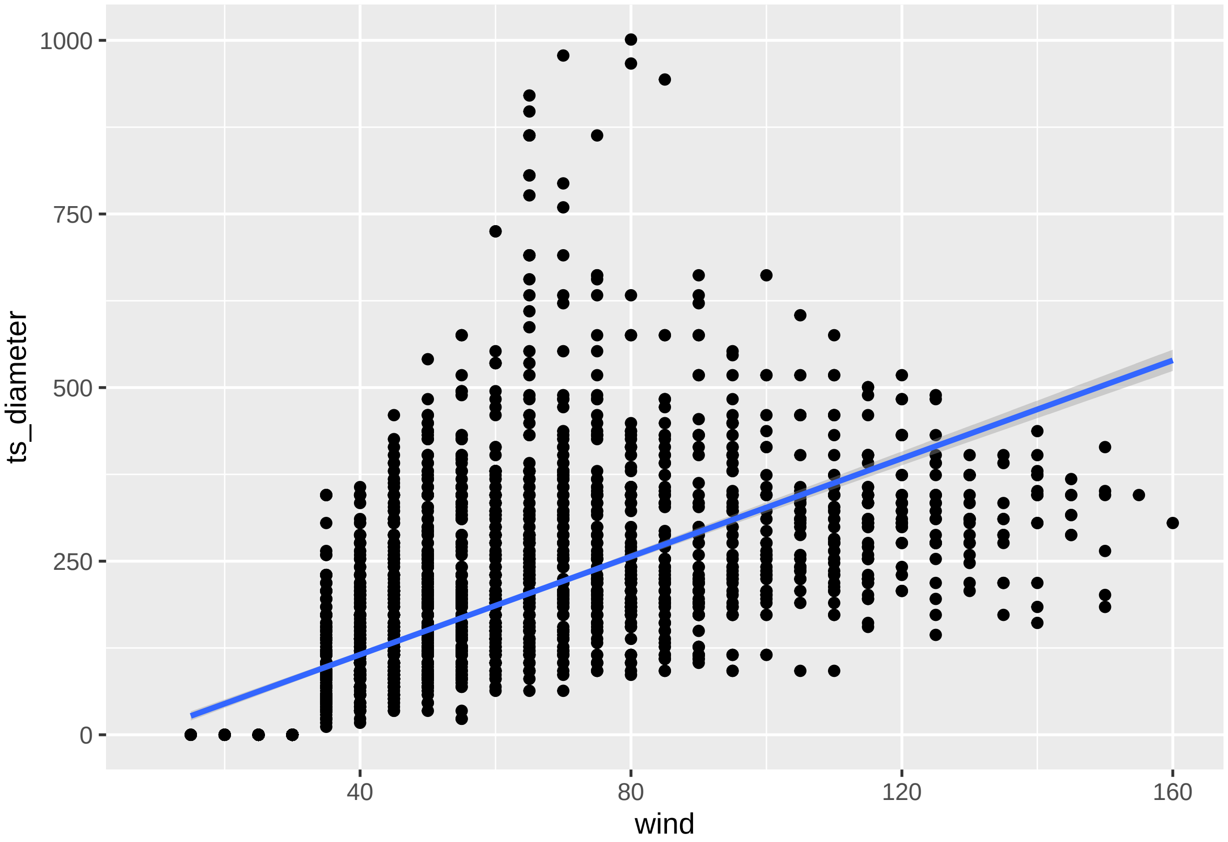


```
ggplot(storms80, aes(x = name, y = pressure)) +
  geom_violin()
```



5) Make a scatterplot of wind (x-axis) and ts_diameter (y-axis), and add a regression line—via geom_smooth().

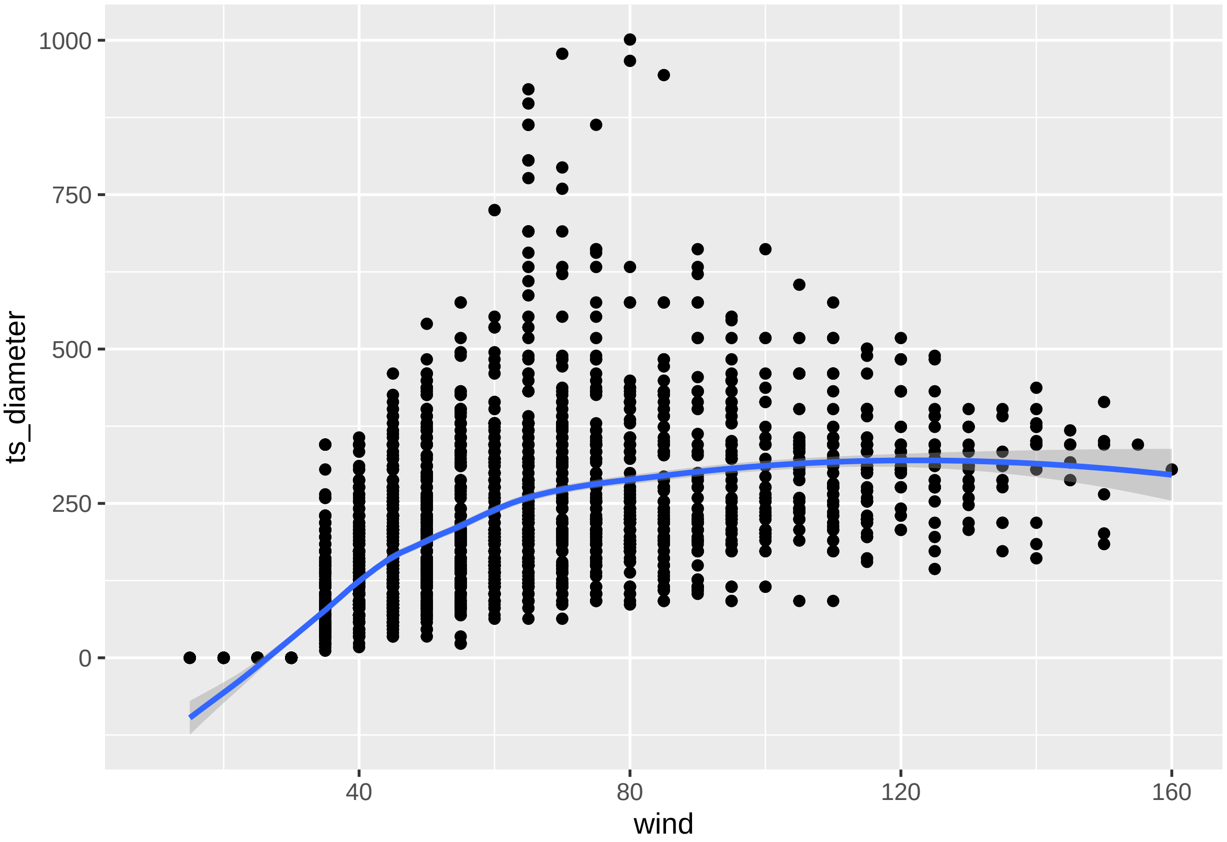
```
ggplot(storms, aes(x = wind, y = ts_diameter)) +
  geom_point(na.rm = TRUE) + # remove missing values from the data
  geom_smooth(method = "lm", na.rm = TRUE)
```



Try geom_smooth() with method = lm to fit a least squares regression line.

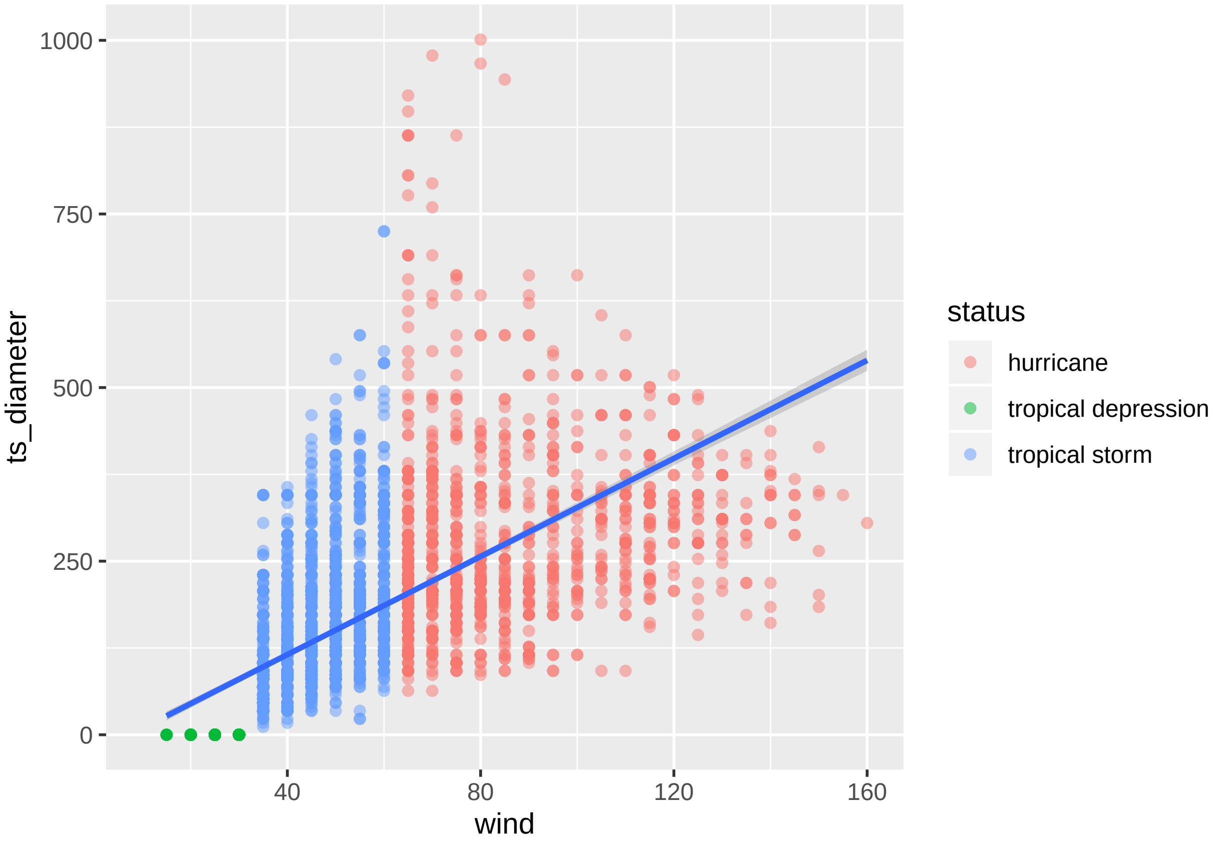
Try geom_smooth() with method = loess to fit a local polynomial regression.

```
ggplot(storms, aes(x = wind, y = ts_diameter)) +
  geom_point(na.rm = TRUE) +
  geom_smooth(method = "loess", na.rm = TRUE)
```



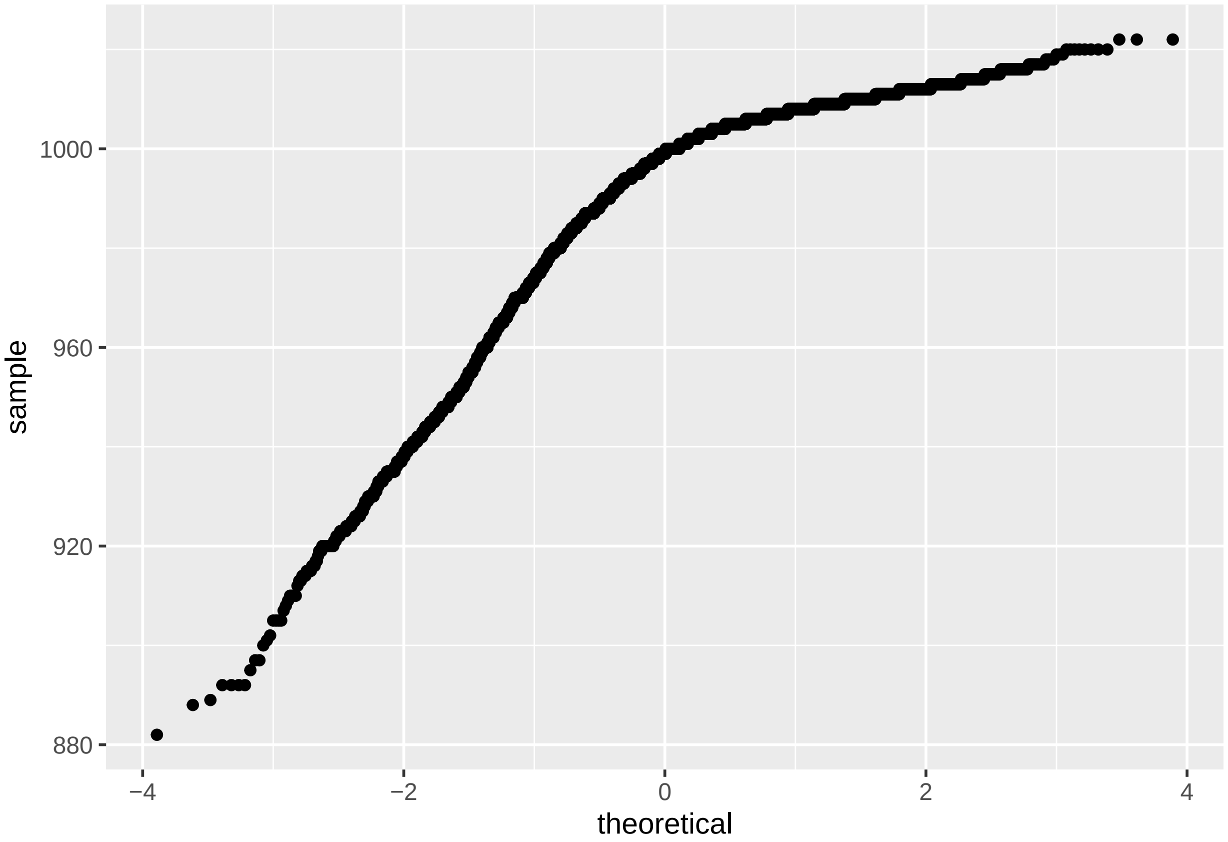
6) Repeat the previous scatterplot of wind (x-axis) and ts_diameter (y-axis), but now use status to color code the points, and use the alpha argument to add some transparency to the dots.

```
ggplot(storms, aes(x = wind, y = ts_diameter)) +
  geom_point(aes(color = status), alpha = 0.5, na.rm = TRUE) +
  geom_smooth(method = "lm", na.rm = TRUE)
```

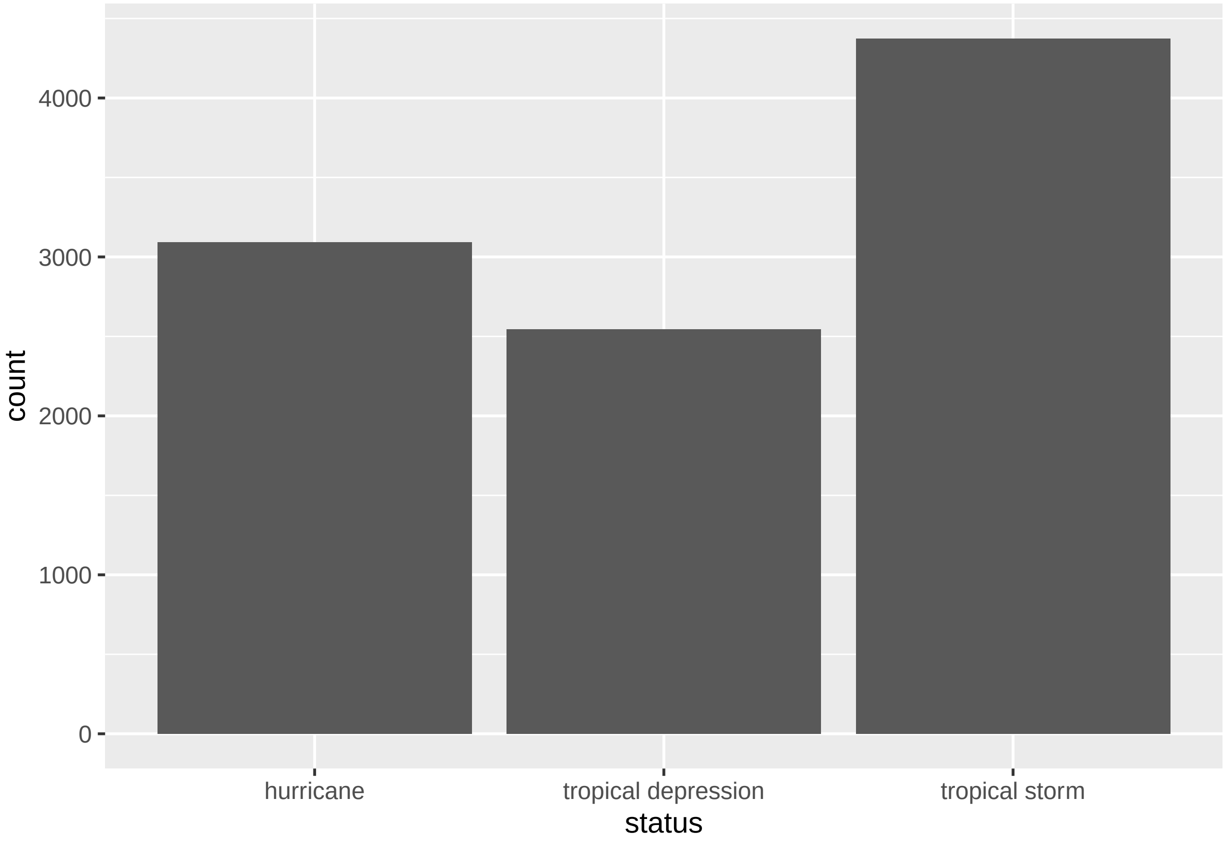


7) Take a look at the cheatsheet of “ggplot2” and make at least 5 more different graphs (e.g. of one variable, of two variables, of three variables).

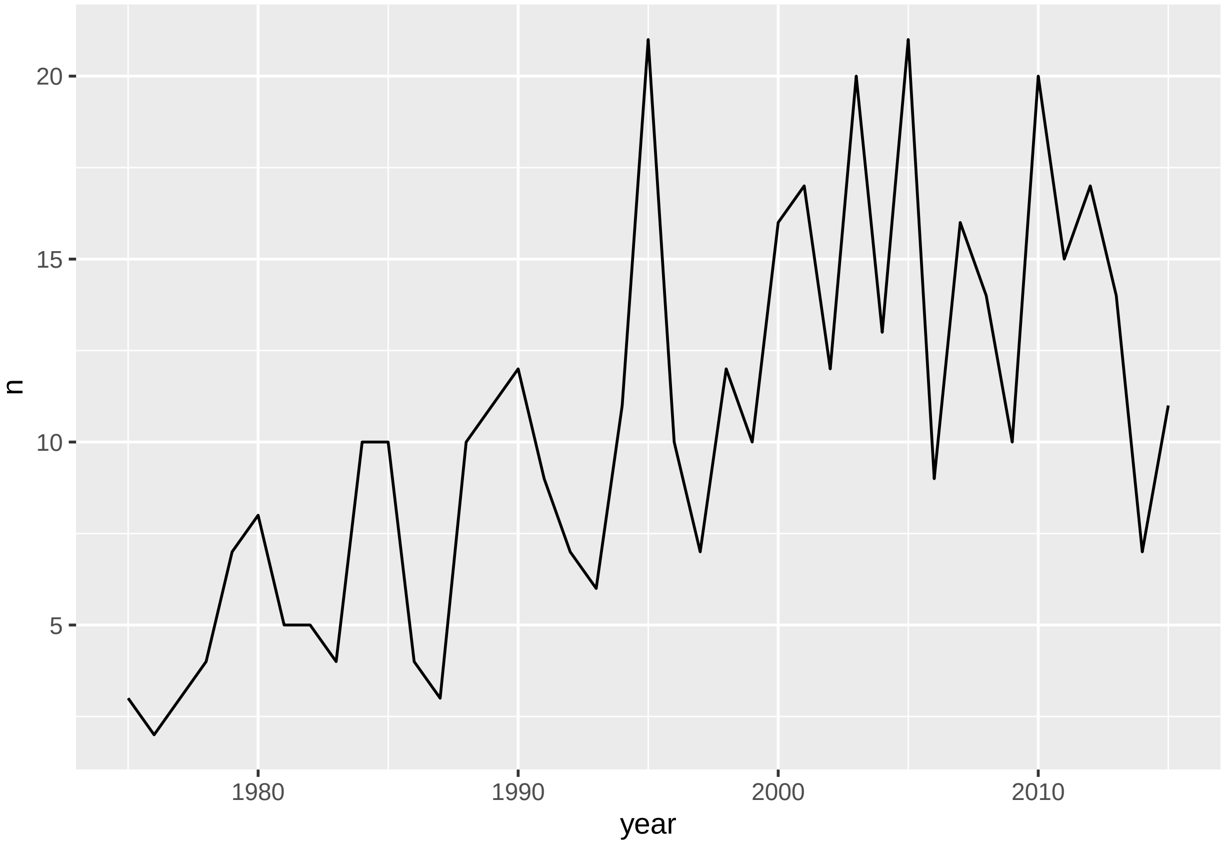
```
# one variable
ggplot(storms) + geom_qq(aes(sample = pressure))
```



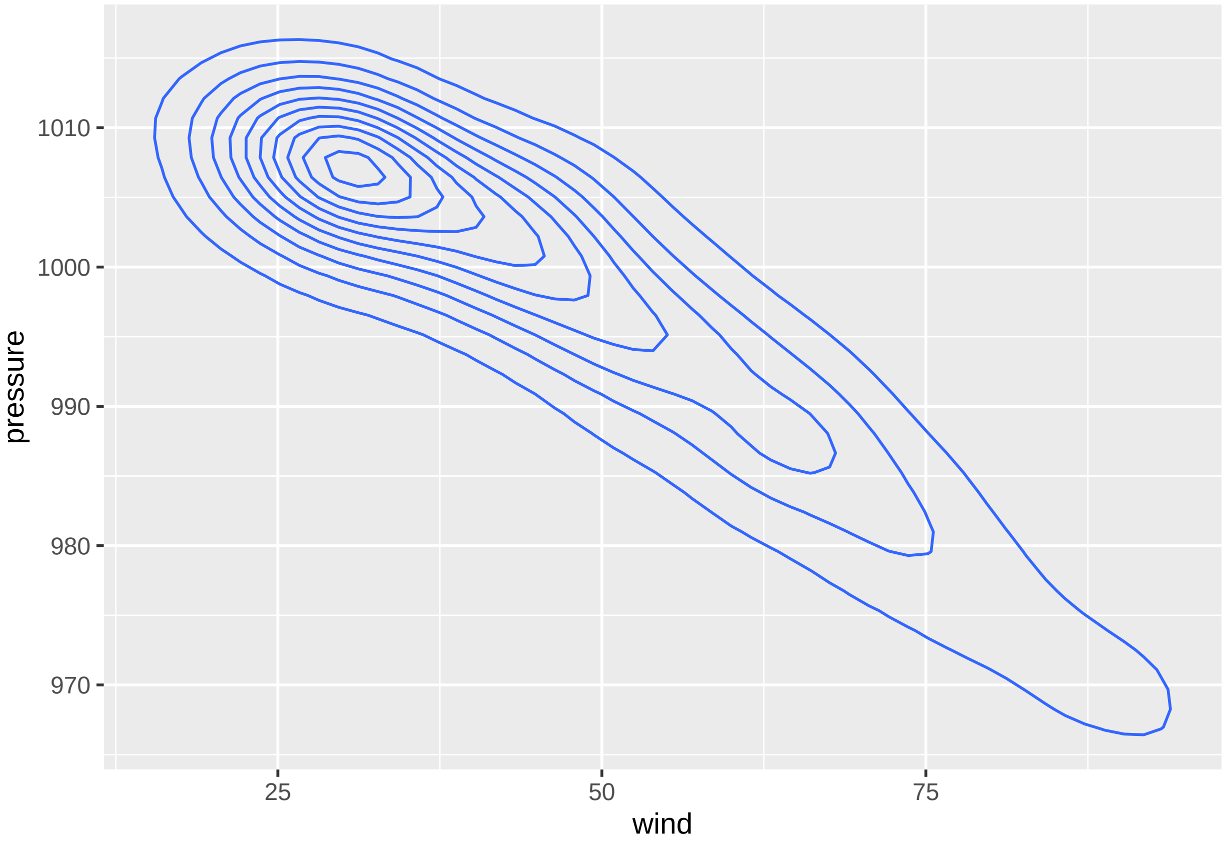
```
ggplot(storms) + geom_bar(aes(x = status))
```



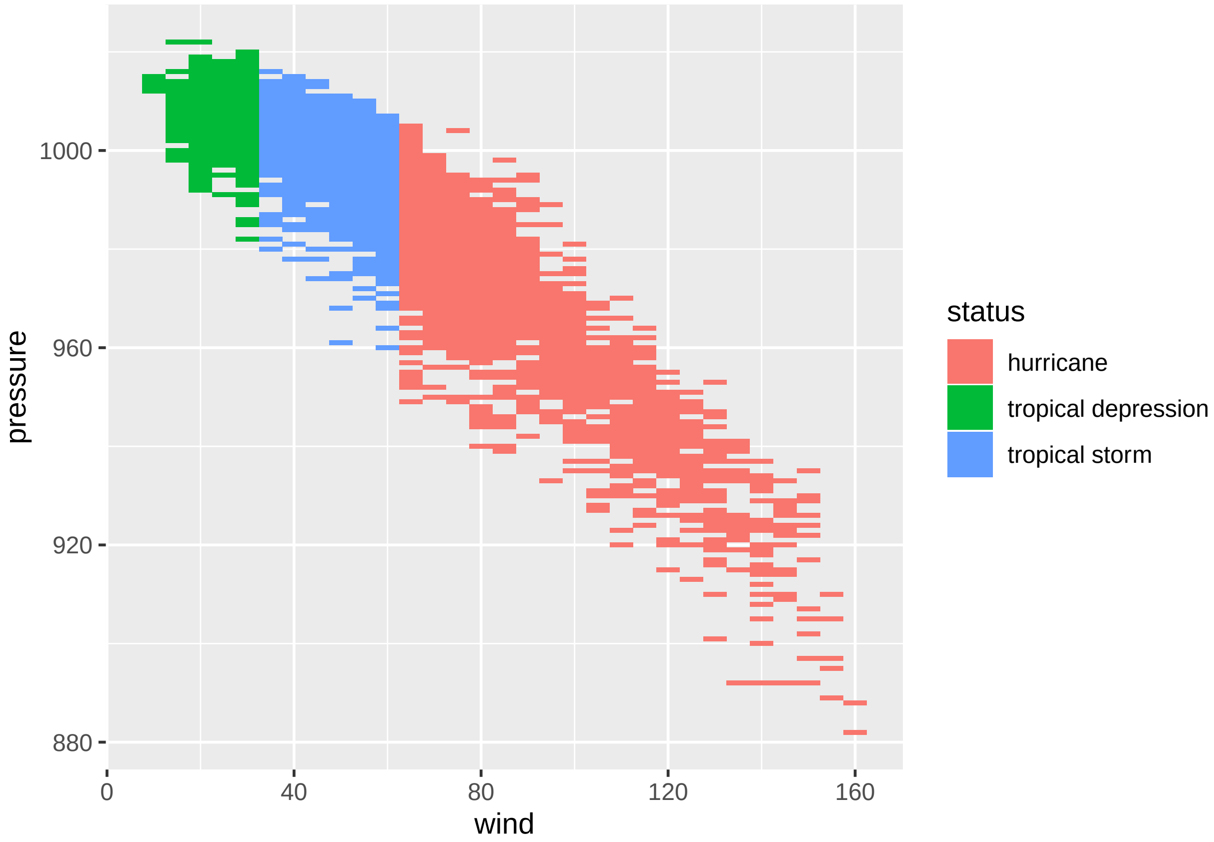
```
# two variables
ggplot(count(storms_year_name), aes(x = year, y = n)) + geom_line()
```



```r
ggplot(storms, aes(x = wind, y = pressure)) + geom_density2d()
```



```
# three variables
ggplot(storms, aes(x = wind, y = pressure)) + geom_tile(aes(fill = status))
```


pressure
1000
960
920
880
0
40
80
120
160
wind
status
hurricane
tropical depression
tropical storm