Transform!

Charlotte Wickham
Assistant Professor, Oregon State University
cwickham@gmail.com

October 2015

Adapted from slides by Hadley Wickham

Import

4

readr
readxl
haven

1

Visualise

ggplot2
ggvis

3

2

Tidy

tidyr

Transform

dplyr

Model

Two table verbs

```
# Motivation: how can we show airport delays on
# a map?  Need to connect to airports dataset

delays <- flights %>%
  group_by(dest) %>%
  summarise(
    arr_delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )
delays <- delays %>%
  left_join(airports, c("dest" = "faa"))
```

Joining datasets

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

?

```
x <- data.frame(
  name = c("John", "Paul", "George", "Ringo", "Stuart", "Pete"),
  instrument = c("guitar", "bass", "guitar", "drums", "bass",
    "drums")
)

y <- data.frame(
  name = c("John", "Paul", "George", "Ringo", "Brian"),
  band = c("TRUE", "TRUE", "TRUE",  "TRUE", "FALSE")
)
```

x

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T

```
inner_join(x, y)
```

x

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Stuart	bass	NA
Pete	drums	NA

```
left_join(x, y)
```

x

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums

```
semi_join(x, y)
```

x

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

name	instrument
Stuart	bass
Pete	drums

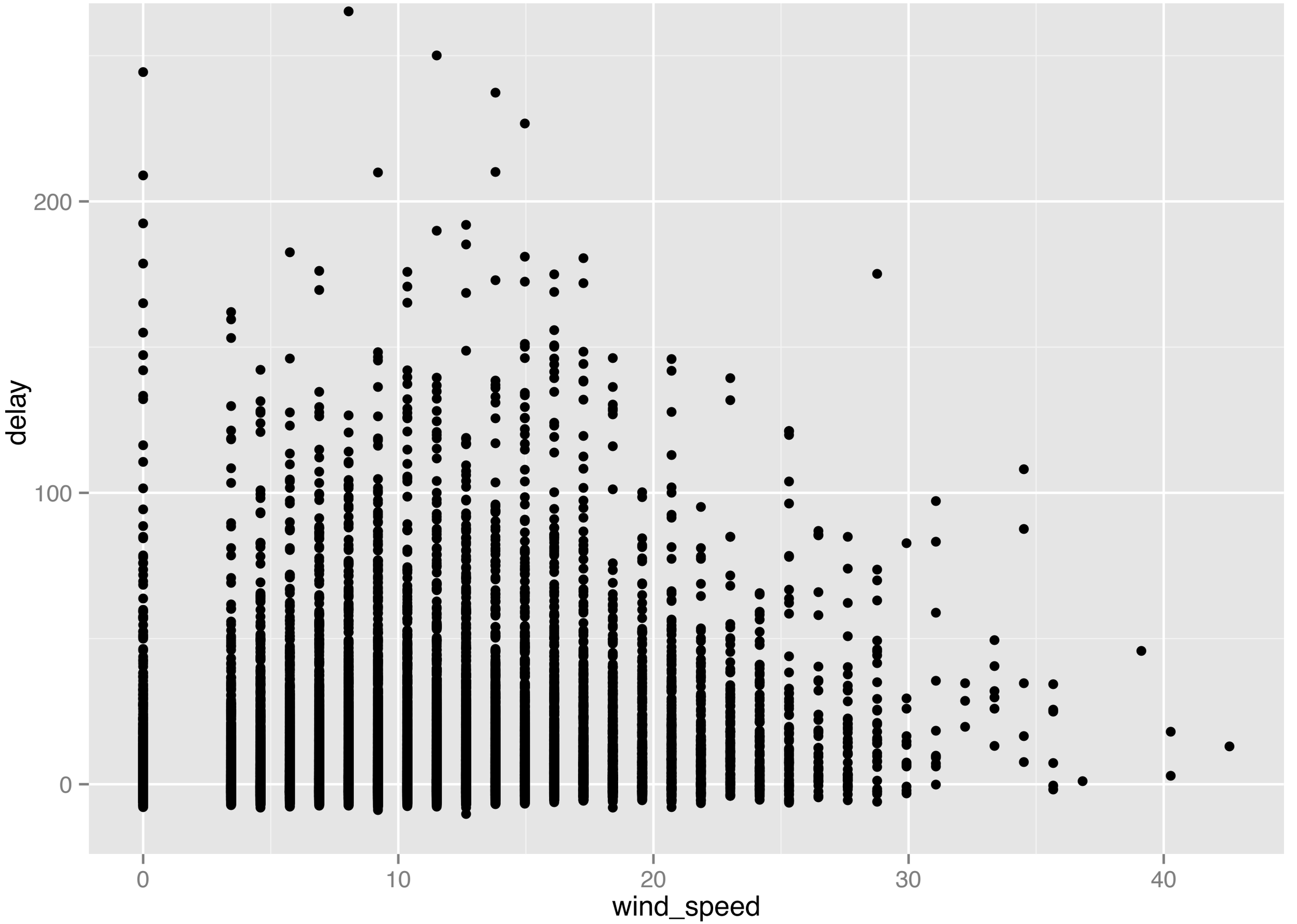
```
anti_join(x, y)
```

Type	Action
inner	Include only rows in **both** x and y
left	Include all of x, and matching rows of y
semi	Include rows of x that match y
anti	Include rows of x that **don't** match y

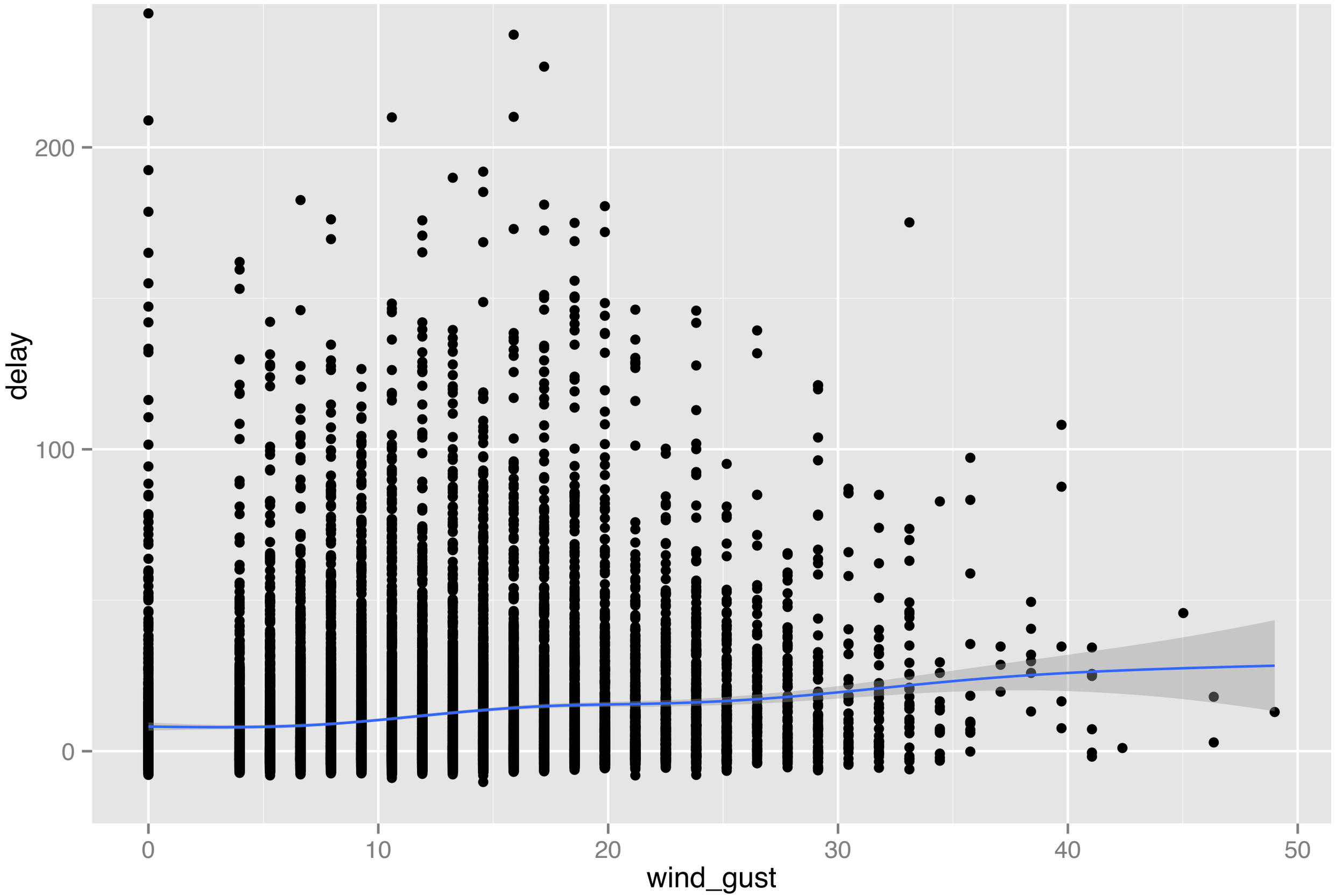
```
# Let's combine hourly delay data with weather
# information

hourly_delay <- flights %>%
  group_by(date, hour) %>%
  filter(!is.na(dep_delay)) %>%
  summarise(
    delay = mean(dep_delay),
    n = n()
  ) %>%
  filter(n > 10)
delay_weather <- hourly_delay %>% left_join(weather)
```

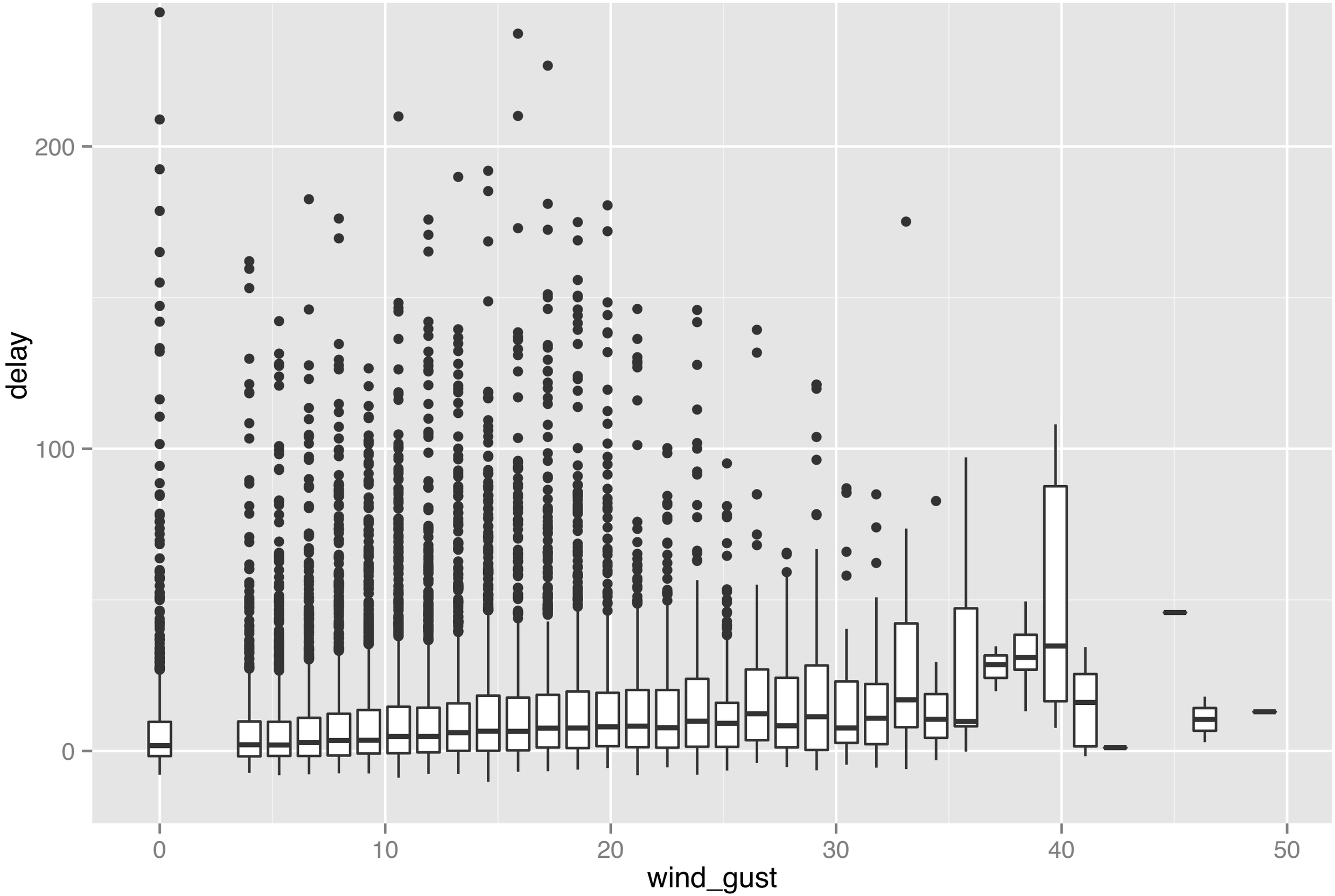
```
qplot(wind_gust, delay, data = delay_weather)
```



```
qplot(wind_gust, delay, data = delay_weather) +
  geom_smooth()
```



```
qplot(wind_gust, delay, data = delay_weather,
      geom = "boxplot", group = wind_gust)
```



Your turn

Which plane (tailnum) flew the most flights?

How much did it rain each month at JFK?

How much did it rain each month for each airports?

Which airport had the highest arrival delays? Is the distance to the airport related to the delay?

Your turn

What weather conditions are associated with delays leaving in NYC?

Use graphics to explore.

Your turn

Are older planes more likely to be delayed? Explore the data and answer with a plot.

(Hint: I'd recommend by starting with some checking of the plane data)

Where next

```
browseVignettes(package = "dplyr")

# Translate plyr to dplyr
http://jimhester.github.io/plyrToDplyr/

# Common questions & answers
http://stackoverflow.com/questions/tagged/dplyr?sort=frequent
```