PM 566 HW 02

AUTHOR
Ziquan ‘Harrison’ Liu

Packages

```r
library(nycflights13)
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
library(ggplot2)
library(lubridate)
```

```

Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```r
library(stringr)
library(maps)
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ forcats 1.0.1     ✔ tibble  3.3.0
✔ purrr   1.1.0     ✔ tidyr   1.3.1
✔ readr   2.1.5     
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
✖ purrr::map()    masks maps::map()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

```r
library(scales)
```

```

Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard

The following object is masked from 'package:readr':

    col_factor
```

```r
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library(knitr)
library(leaflet)
library(forcats)
library(tidytext)
library(magrittr)
```

```

Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

    set_names

The following object is masked from 'package:tidyr':

    extract
```

```r
library(rstatix)
```

```

Attaching package: 'rstatix'

The following object is masked from 'package:janitor':

    make_clean_names
```

```
The following object is masked from 'package:stats':

    filter
```

```r
library(tidyr)
library(patchwork)  # to collect guides into one legend
library(hexbin)
library(ggcorrplot)
```

```

Attaching package: 'ggcorrplot'

The following object is masked from 'package:rstatix':

    cor_pmat
```

Check on all description of dataset

```r
summary(flights)
```

```
      year          month             day            dep_time    sched_dep_time
 Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1   Min.   : 106  
 1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907   1st Qu.: 906  
 Median :2013   Median : 7.000   Median :16.00   Median :1401   Median :1359  
 Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349   Mean   :1344  
 3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744   3rd Qu.:1729  
 Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400   Max.   :2359  
                                                 NA's   :8255                 
   dep_delay          arr_time    sched_arr_time   arr_delay       
 Min.   : -43.00   Min.   :   1   Min.   :   1   Min.   : -86.000  
 1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124   1st Qu.: -17.000  
 Median :  -2.00   Median :1535   Median :1556   Median :  -5.000  
 Mean   :  12.64   Mean   :1502   Mean   :1536   Mean   :   6.895  
 3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945   3rd Qu.:  14.000  
 Max.   :1301.00   Max.   :2400   Max.   :2359   Max.   :1272.000  
 NA's   :8255      NA's   :8713                  NA's   :9430      
   carrier              flight       tailnum             origin         
 Length:336776      Min.   :   1   Length:336776      Length:336776     
 Class :character   1st Qu.: 553   Class :character   Class :character  
 Mode  :character   Median :1496   Mode  :character   Mode  :character  
                    Mean   :1972                                        
                    3rd Qu.:3465                                        
                    Max.   :8500                                        
                                                                        
     dest              air_time        distance         hour      
 Length:336776      Min.   : 20.0   Min.   :  17   Min.   : 1.00  
 Class :character   1st Qu.: 82.0   1st Qu.: 502   1st Qu.: 9.00  
 Mode  :character   Median :129.0   Median : 872   Median :13.00  
```

```
                   Mean   :150.7   Mean   :1040   Mean   :13.18  
                   3rd Qu.:192.0   3rd Qu.:1389   3rd Qu.:17.00  
                   Max.   :695.0   Max.   :4983   Max.   :23.00  
                   NA's   :9430                                  
    minute        time_hour                    
Min.   : 0.00   Min.   :2013-01-01 05:00:00  
1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00  
Median :29.00   Median :2013-07-03 10:00:00  
Mean   :26.23   Mean   :2013-07-03 05:22:54  
3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00  
Max.   :59.00   Max.   :2013-12-31 23:00:00  
```

```r
summary(airlines)
```

```
  carrier              name          
Length:16          Length:16          
Class :character   Class :character  
Mode  :character   Mode  :character  
```

```r
summary(airports)
```

```
    faa                name                lat             lon         
Length:1458        Length:1458        Min.   :19.72   Min.   :-176.65  
Class :character   Class :character   1st Qu.:34.26   1st Qu.:-119.19  
Mode  :character   Mode  :character   Median :40.09   Median : -94.66  
                                      Mean   :41.65   Mean   :-103.39  
                                      3rd Qu.:45.07   3rd Qu.: -82.52  
                                      Max.   :72.27   Max.   : 174.11  
     alt               tz               dst                tzone          
Min.   : -54.00   Min.   :-10.000   Length:1458        Length:1458       
1st Qu.:  70.25   1st Qu.: -8.000   Class :character   Class :character  
Median : 473.00   Median : -6.000   Mode  :character   Mode  :character  
Mean   :1001.42   Mean   : -6.519                                        
3rd Qu.:1062.50   3rd Qu.: -5.000                                        
Max.   :9078.00   Max.   :  8.000                                        
```

```r
summary(planes)
```

```
  tailnum             year          type           manufacturer      
Length:3322        Min.   :1956   Length:3322        Length:3322       
Class :character   1st Qu.:1997   Class :character   Class :character  
Mode  :character   Median :2001   Mode  :character   Mode  :character  
                   Mean   :2000                                        
                   3rd Qu.:2005                                        
                   Max.   :2013                                        
                   NA's   :70                                          
   model              engines          seats            speed      
Length:3322        Min.   :1.000   Min.   :  2.0   Min.   : 90.0  
```

```
 Class :character   1st Qu.:2.000   1st Qu.:140.0   1st Qu.:107.5  
 Mode  :character   Median :2.000   Median :149.0   Median :162.0  
                    Mean   :1.995   Mean   :154.3   Mean   :236.8  
                    3rd Qu.:2.000   3rd Qu.:182.0   3rd Qu.:432.0  
                    Max.   :4.000   Max.   :450.0   Max.   :432.0  
                                                    NA's   :3299   
    engine         
 Length:3322       
 Class :character  
 Mode  :character  
```

```
summary(weather)
```

```
    origin               year          month             day       
 Length:26115       Min.   :2013   Min.   : 1.000   Min.   : 1.00  
 Class :character   1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00  
 Mode  :character   Median :2013   Median : 7.000   Median :16.00  
                    Mean   :2013   Mean   : 6.504   Mean   :15.68  
                    3rd Qu.:2013   3rd Qu.: 9.000   3rd Qu.:23.00  
                    Max.   :2013   Max.   :12.000   Max.   :31.00  
                                                                   
      hour            temp             dewp           humid       
 Min.   : 0.00   Min.   : 10.94   Min.   :-9.94   Min.   : 12.74  
 1st Qu.: 6.00   1st Qu.: 39.92   1st Qu.:26.06   1st Qu.: 47.05  
 Median :11.00   Median : 55.40   Median :42.08   Median : 61.79  
 Mean   :11.49   Mean   : 55.26   Mean   :41.44   Mean   : 62.53  
 3rd Qu.:17.00   3rd Qu.: 69.98   3rd Qu.:57.92   3rd Qu.: 78.79  
 Max.   :23.00   Max.   :100.04   Max.   :78.08   Max.   :100.00  
                 NA's   :1        NA's   :1       NA's   :1       
    wind_dir       wind_speed         wind_gust         precip        
 Min.   :  0.0   Min.   :   0.000   Min.   :16.11   Min.   :0.000000  
 1st Qu.:120.0   1st Qu.:   6.905   1st Qu.:20.71   1st Qu.:0.000000  
 Median :220.0   Median :  10.357   Median :24.17   Median :0.000000  
 Mean   :199.8   Mean   :  10.517   Mean   :25.49   Mean   :0.004469  
 3rd Qu.:290.0   3rd Qu.:  13.809   3rd Qu.:28.77   3rd Qu.:0.000000  
 Max.   :360.0   Max.   :1048.361   Max.   :66.75   Max.   :1.210000  
 NA's   :460     NA's   :4          NA's   :20778                     
    pressure          visib          time_hour                     
 Min.   : 983.8   Min.   : 0.000   Min.   :2013-01-01 01:00:00  
 1st Qu.:1012.9   1st Qu.:10.000   1st Qu.:2013-04-01 21:30:00  
 Median :1017.6   Median :10.000   Median :2013-07-01 14:00:00  
 Mean   :1017.9   Mean   : 9.255   Mean   :2013-07-01 18:26:37  
 3rd Qu.:1023.0   3rd Qu.:10.000   3rd Qu.:2013-09-30 13:00:00  
 Max.   :1042.1   Max.   :10.000   Max.   :2013-12-30 18:00:00  
 NA's   :2729                                                   
```

standardize time

```r
# helper: convert HHMM integer time (e.g., 517) to hour-of-day on [0,24)
to_hour <- function(x) ifelse(is.na(x), NA_real_, (x %/% 100) %% 24 + (x %% 100)/60)

# helper: map hour to part-of-day
part_of_day <- function(hour) {
  cut(hour,
      breaks = c(0, 6, 12, 18, 24),
      labels = c("early morning", "morning", "afternoon", "evening"),
      right = FALSE, include.lowest = TRUE)
}
```

Question 1

```r
top10_dest <- flights %>%
  count(dest, sort = TRUE, name = "n_flights") %>%
  slice_head(n = 10)
top10_dest
```

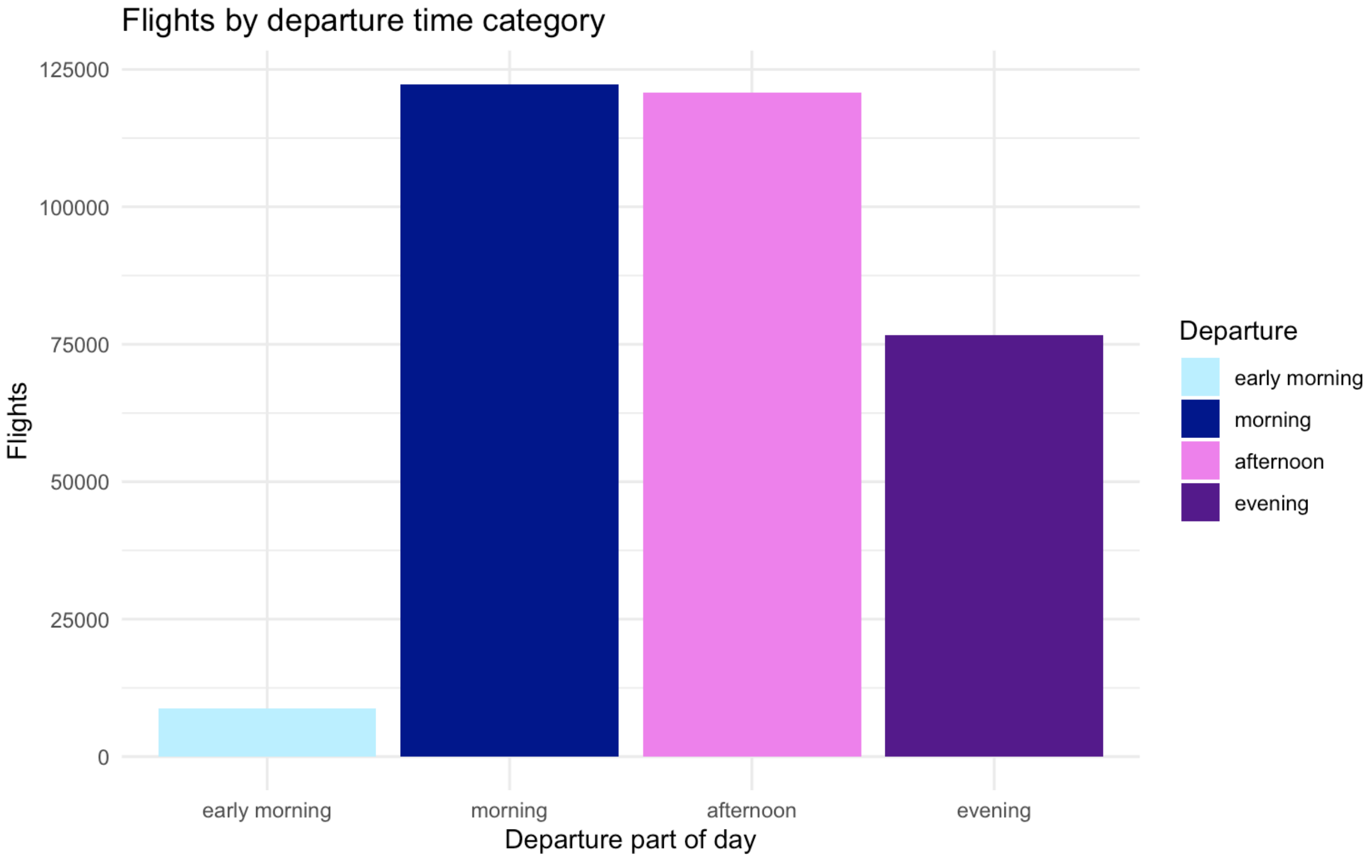
```
# A tibble: 10 × 2
   dest  n_flights
   <chr>     <int>
 1 ORD       17283
 2 ATL       17215
 3 LAX       16174
 4 BOS       15508
 5 MCO       14082
 6 CLT       14064
 7 SFO       13331
 8 FLL       12055
 9 MIA       11728
10 DCA        9705
```

Based on the result above, The top 10 most popular destinations with number of flights are as follows: ORD with 17283 flights, ATL with 17215 flights, LAX with 16174 flights, BOS with 15508 flights, MCO with 14082 flights, CLT with 14064 flights, SFO with 13331 flights, FLL with 12055 flights, MIA with 11728 flights, and DCA with 9705 flights.

Question 2

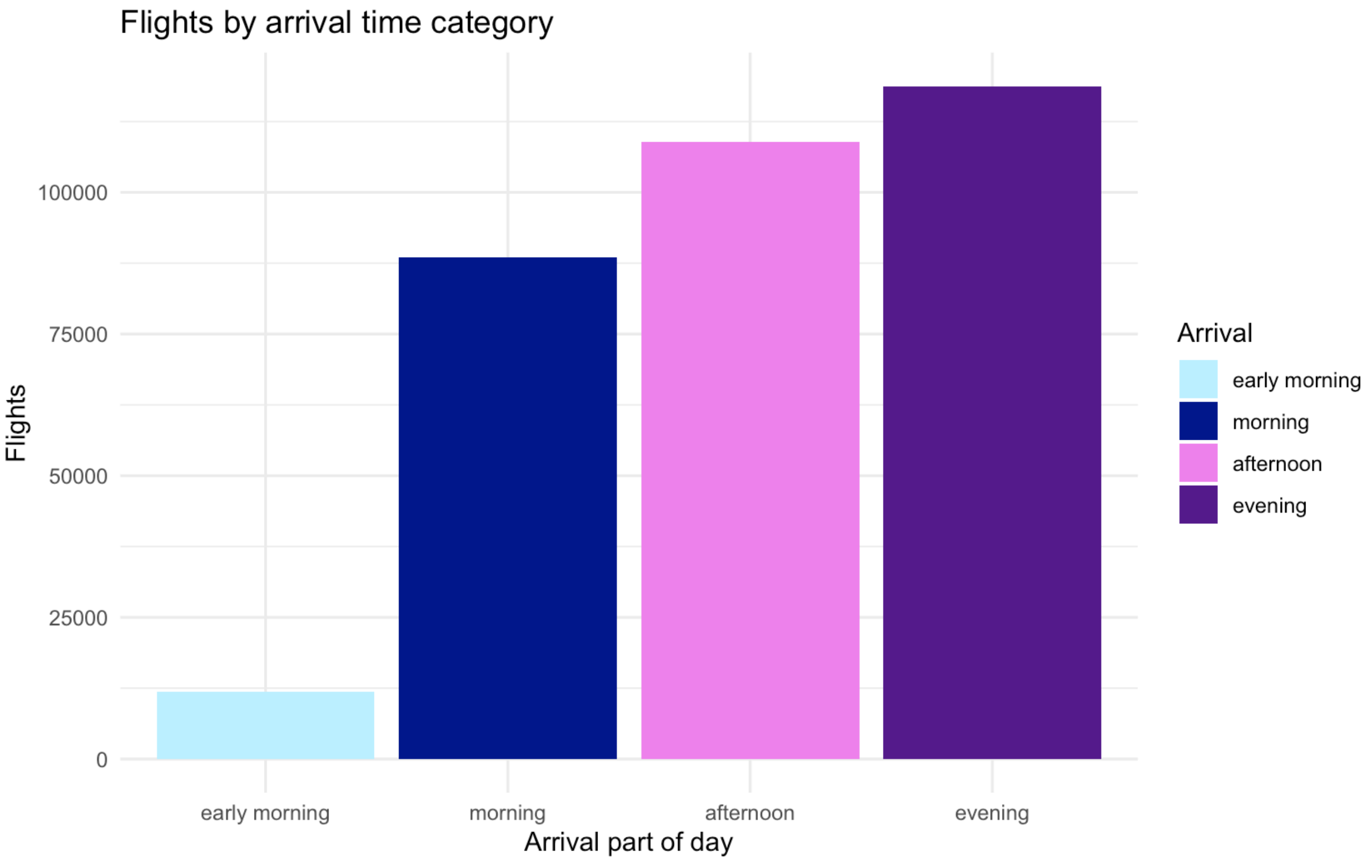
```
flights2 <- flights %>%
  mutate(
    dep_hour = to_hour(dep_time),
    arr_hour = to_hour(arr_time),
    dep_part = part_of_day(dep_hour),
    arr_part = part_of_day(arr_hour)
  )

# barplots
## select corlor
pal <- c(
  "early morning" = "lightblue1",
  "morning"       = "darkblue",
  "afternoon"     = "violet",
  "evening"       = "purple4"
)

ggplot(flights2, aes(x = dep_part, fill = dep_part)) +
  geom_bar() +
  scale_x_discrete(na.translate = FALSE) +         # mute NA category
  scale_fill_manual(values = pal, na.translate = FALSE) +
  labs(x = "Departure part of day", y = "Flights",
       title = "Flights by departure time category") +
  guides(fill = guide_legend(title = "Departure")) +
  theme_minimal()
```

```
Warning: Removed 8255 rows containing non-finite outside the scale range
(`stat_count()`).
```



```r
ggplot(flights2, aes(x = arr_part, fill = arr_part)) +
  geom_bar() +
  scale_x_discrete(na.translate = FALSE) +
  scale_fill_manual(values = pal, na.translate = FALSE) +
  labs(x = "Arrival part of day", y = "Flights",
       title = "Flights by arrival time category") +
  guides(fill = guide_legend(title = "Arrival")) +
  theme_minimal()
```

```
Warning: Removed 8713 rows containing non-finite outside the scale range
(`stat_count()`).
```



```
valid <- flights2 %>%                                   # flights2 has dep_part & arr_par
  filter(!is.na(dep_part), !is.na(arr_part))            # keep only rows with both parts
# generate red_eye
red_eye <- valid %>%                                    # work on the valid data
  mutate(is_redeye =                                     # create a logical flag: is this
           dep_part %in% c("afternoon","evening") &      # TRUE if it departs in the afte
           arr_part %in% c("early morning","morning")) %>% # AND TRUE if arrives in early
  summarise(                                             # collapse into a 1-row summary
    n = n(),                                             # denominator: number of flights
    n_redeye = sum(is_redeye),                           # count of red-eye flights (TRUE
    pct_redeye = 100 * n_redeye / n                      # percentage of red-eye flights
  )
red_eye                                                  # print the summary table
```

```
# A tibble: 1 × 3
       n n_redeye pct_redeye
   <int>    <int>      <dbl>
1 328063    10754       3.28
```

Based on the result above, after rmoved NA, the barplots are above. The percentage of flights were "red eye" flights was about 3.28%.

Question 3

```
tail_carriers <- flights %>%
  filter(!is.na(tailnum), tailnum != "", !is.na(carrier)) %>%         # keep rows with a
  distinct(tailnum, carrier) %>%                                       # reduce to unique
  left_join(airlines, by = "carrier")                                  # attach full airl
# count distinct carriers per plane, keep those with >1
multi_airline_planes <- tail_carriers %>%                              # work on the uniq
  group_by(tailnum) %>%                                                # one summary per
  summarise(
    n_airlines = n_distinct(carrier),                                  # how many differe
    airlines   = paste(sort(unique(name)), collapse = ", "),           # list those carri
    .groups    = "drop"                                                # return an ungrou
  ) %>%
  filter(n_airlines > 1) %>%                                           # keep only planes
  arrange(desc(n_airlines), tailnum)                                   # order by most ca
# how many such planes?
n_multi_planes <- nrow(multi_airline_planes)                           # count how many s
n_multi_planes                                                         # print count
```

```
[1] 17
```

```
multi_airline_planes                                                   # print detailed t
```

```
# A tibble: 17 × 3
   tailnum n_airlines airlines
   <chr>        <int> <chr>
 1 N146PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 2 N153PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 3 N176PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 4 N181PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 5 N197PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 6 N200PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 7 N228PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 8 N232PQ           2 Endeavor Air Inc., ExpressJet Airlines Inc.
 9 N933AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
10 N935AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
11 N977AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
12 N978AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
13 N979AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
14 N981AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
15 N989AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
16 N990AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
17 N994AT           2 AirTran Airways Corporation, Delta Air Lines Inc.
```

Based on the result above, there are 17 in total planes that flew for multiple airlines. Such airlines were: Endeavor Air Inc., ExpressJet

Airlines Inc., and AirTran Airways Corporation, Delta Air Lines Inc..

Question 4

```
table(weather$origin)
```

```
 EWR  JFK  LGA 
8703 8706 8706 
```

```
table(airports$faa)
```

```
04G 06A 06C 06N 09J 0A9 0G6 0G7 0P2 0S9 0W3 10C 17G 19A 1A3 1B9 1C9 1CS 1G3 1G4 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
1H2 1OH 1RL 23M 24C 24J 25D 29D 2A0 2B2 2G2 2G9 2H0 2J9 369 36U 38W 3D2 3G3 3G4 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
3J1 3W2 40J 41N 47A 49A 49X 4A4 4A7 4A9 4B8 4G0 4G2 4G4 4I7 4U9 52A 54J 55J 55S 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
57C 5B2 60J 6A2 6J4 6K8 6S0 6S2 6Y8 70J 70N 7A4 7D9 7N7 8M8 93C 99N 9A1 9A5 9G1 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
A39 A50 AAF AAP ABE ABI ABL ABQ ABR ABY ACJ ACK ACT ACV ACY ADK ADM ADQ ADS ADW 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
AET AEX AFE AFW AGC AGN AGS AHN AIA AIK AIN AIZ AKB AKC AKI AKK AKN AKP ALB ALI 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
ALM ALO ALS ALW ALX ALZ AMA ANB ANC AND ANI ANN ANP ANQ ANV AOH AOO AOS APA APC 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
APF APG APN AQC ARA ARB ARC ART ARV ASE ASH AST ATK ATL ATT ATW ATY AUG AUK AUO 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
AUS AUW AVL AVO AVP AVW AVX AZA AZO BAB BAD BAF BBX BCE BCT BDE BDL BDR BEC BED 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
BEH BET BFD BFF BFI BFL BFM BFP BFT BGE BGM BGR BHB BHM BID BIF BIG BIL BIS BIV 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
BIX BJC BJI BKC BKD BKF BKG BKH BKL BKW BKX BLD BLF BLH BLI BLV BMC BMG BMI BMT 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
BMX BNA BOI BOS BOW BPT BQK BRD BRL BRO BRW BSF BTI BTM BTR BTT BTV BUF BUR BUU 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
BUY BVY BWD BWG BWI BXK BXS BYH BYS BYW BZN C02 C16 C47 C65 C89 C91 CAE CAK CAR 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
CBE CBM CCO CCR CDB CDC CDI CDK CDN CDR CDS CDV CDW CEC CEF CEM CEU CEW CEZ CFD 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
CGA CGC CGF CGI CGX CGZ CHA CHI CHO CHS CHU CIC CID CIK CIL CIU CKB CKD CKF CKV 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
CLC CLD CLE CLL CLM CLS CLT CLW CMH CMI CMX CNM CNW CNY COD COE COF COI CON COS 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
COT COU CPR CPS CRE CRP CRW CSG CTB CTH CTJ CTY CVG CVN CVO CVS CVX CWA CWI CWT 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
CXF CXL CXO CXY CYF CYM CYS CYT CZF CZG CZN DAB DAL DAW DAY DBN DBQ DCA DDC DEC 
```

```
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
DEN DET DFW DGL DHB DHN DHT DIK DKB DKK DKX DLF DLG DLH DLL DMA DNL DNN DNV DOV 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
DPA DQH DRG DRI DRM DRO DRT DSM DTA DTS DTW DUC DUG DUJ DUT DVL DVT DWA DWH DWS 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
DXR DYS E25 E51 E55 E63 E91 EAA EAR EAT EAU ECA ECG ECP EDF EDW EEK EEN EET EFD 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
EGA EGE EGT EGV EGX EHM EIL EKI EKN EKO EKY ELD ELI ELM ELP ELV ELY EMK EMP ENA 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
END ENV ENW EOK EPM EQY ERI ERV ERY ESC ESD ESF ESN EUF EUG EVV EVW EWB EWK EWN 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
EWR EXI EYW F57 FAF FAI FAR FAT FAY FBG FBK FBR FBS FCA FCM FCS FDW FDY FFA FFC 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
FFO FFT FFZ FHU FIT FKL FLD FLG FLL FLO FLV FME FMH FMN FMY FNL FNR FNT FOD FOE 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
FOK FRD FRI FRN FRP FSD FSI FSM FST FTK FTW FTY FUL FWA FXE FYU FYV FZG FZI GAD 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
GAI GAL GAM GBN GCC GCK GCN GCW GDV GDW GED GEG GEU GFK GFL GGE GGG GGW GHG GIF 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
GJT GKN GKY GLD GLH GLS GLV GNT GNU GNV GON GPT GPZ GQQ GRB GRF GRI GRK GRM GRR 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
GSB GSO GSP GST GTB GTF GTR GTU GUC GUP GUS GVL GVQ GVT GWO GYY HBG HBR HCC HCR 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
HDH HDI HDN HDO HFD HGR HHH HHI HHR HIB HIF HII HIO HKB HKY HLG HLN HLR HMN HNH 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
HNL HNM HNS HOB HOM HON HOP HOT HOU HPB HPN HQM HQU HRL HRO HRT HSH HSL HST HSV 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
HTL HTS HUA HUF HUL HUS HUT HVN HVR HWD HWO HXD HYA HYG HYL HYS HZL IAB IAD IAG 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
IAH IAN ICT ICY IDA IDL IFP IGG IGM IGQ IJD IKK IKO IKR IKV ILG ILI ILM ILN IMM 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
IMT IND INJ INK INL INS INT INW IOW IPL IPT IRC IRK ISM ISN ISO ISP ISW ITH ITO 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
IWD IWS IYK IZG JAC JAN JAX JBR JCI JEF JES JFK JGC JHM JHW JKA JLN JMS JNU JOT 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
JRA JRB JST JVL JXN JYL JYO JZP K03 K27 K83 KAE KAL KBC KBW KCC KCL KCQ KEH KEK 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
KFP KGK KGX KKA KKB KKH KLG KLL KLN KLS KLW KMO KMY KNW KOA KOT KOY KOZ KPB KPC 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
KPN KPR KPV KPY KQA KSM KTB KTN KTS KUK KVC KVL KWK KWN KWP KWT KYK KYU KZB L06 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
L35 L52 LAA LAF LAL LAM LAN LAR LAS LAW LAX LBB LBE LBF LBL LBT LCH LCK LCQ LDJ 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
LEB LEW LEX LFI LFK LFT LGA LGB LGC LGU LHD LHM LHV LHX LIH LIT LIV LKE LKK LKP 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
LMT LNA LNK LNN LNR LNS LNY LOT LOU LOZ LPC LPR LPS LRD LRF LRO LRU LSE LSF LSV 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
LTS LUF LUK LUP LUR LVK LVM LVS LWA LWB LWC LWM LWS LWT LXY LYH LYU LZU M94 MAE 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
MAF MBL MBS MCC MCD MCE MCF MCG MCI MCK MCL MCN MCO MCW MDT MDW ME5 MEI MEM MER 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
```

```
MFD MFE MFI MFR MGC MGE MGJ MGM MGR MGW MGY MHK MHM MHR MHT MHV MIA MIB MIC MIE 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
MIV MKC MKE MKG MKK MKL MKO MLB MLC MLD MLI MLJ MLL MLS MLT MLU MLY MMH MMI MMU 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
MMV MNM MNT MNZ MOB MOD MOT MOU MPB MPI MPV MQB MQI MQT MRB MRI MRK MRN MRY MSL 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
MSN MSO MSP MSS MSY MTC MTH MTJ MTM MTN MUE MUI MUO MVL MVY MWA MWC MWH MWL MWM 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
MXF MXY MYF MYL MYR MYU MYV MZJ N53 N69 N87 NBG NBU NCN NEL NEW NFL NGF NGP NGU 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
NGZ NHK NIB NID NIP NJK NKT NKX NLC NLG NME NMM NNL NOW NPA NPZ NQA NQI NQX NSE 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
NTD NTU NUI NUL NUP NUQ NUW NXP NXX NY9 NYC NYG NZC NZJ NZY O03 O27 OAJ OAK OAR 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
OBE OBU OCA OCF OEB OFF OGG OGS OJC OKC OLF OLH OLM OLS OLT OLV OMA OME OMN ONH 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
ONP ONT OOK OPF OQN OQU ORD ORF ORH ORI ORL ORT ORV OSC OSH OSU OTH OTS OTZ OWB 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
OWD OXC OXD OXR OZA P08 P52 PAE PAH PAM PAO PAQ PBF PBG PBI PBV PBX PCW PCZ PDB 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
PDK PDT PDX PEC PEQ PFN PGA PGD PGV PHD PHF PHK PHL PHN PHO PHX PIA PIB PIE PIH 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
PIM PIP PIR PIT PIZ PKB PLN PMB PMD PML PMP PNC PNE PNM PNS POB POC POE POF PPC 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
PPV PQI PQS PRC PRL PSC PSG PSM PSP PSX PTA PTB PTH PTK PTU PUB PUC PUW PVC PVD 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
PVU PWK PWM PWT PYM PYP R49 RAC RAL RAP RBD RBK RBM RBN RBY RCA RCE RCZ RDD RDG 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
RDM RDR RDU RDV REI RFD RHI RIC RID RIF RIL RIR RIU RIV RIW RKD RKH RKP RKS RME 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
RMG RMP RMY RND RNM RNO RNT ROA ROC ROW RSH RSJ RST RSW RUT RVS RWI RWL RYY S30 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
S40 S46 SAA SAC SAD SAF SAN SAT SAV SBA SBD SBM SBN SBO SBP SBS SBY SCC SCE SCH 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
SCK SCM SDC SDF SDM SDP SDX SDY SEA SEE SEF SEM SES SFB SFF SFM SFO SFZ SGF SGH 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
SGJ SGR SGU SGY SHD SHG SHH SHR SHV SHX SIK SIT SJC SJT SKA SKF SKK SKY SLC SLE 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
SLK SLN SLQ SMD SME SMF SMK SMN SMO SMX SNA SNP SNY SOP SOW SPB SPF SPG SPI SPS 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
SPW SPZ SQL SRQ SRR SRV SSC SSI STC STE STG STJ STK STL STS SUA SUE SUN SUS SUU 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
SUX SVA SVC SVH SVN SVW SWD SWF SXP SXQ SYA SYB SYR SZL TAL TAN TBN TCC TCL TCM 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
TCS TCT TEB TEK TEX TIK TIW TIX TKA TKE TKF TKI TLA TLH TLJ TLT TMA TMB TNC TNK 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
TNT TNX TOA TOC TOG TOL TOP TPA TPL TRI TRM TSS TTD TTN TUL TUP TUS TVC TVF TVI 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
TVL TWA TWD TWF TXK TYE TYR TYS TZR U76 UCA UDD UDG UES UGN UIN UMP UNK UPP UST 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
UT3 UTM UTO UUK UUU UVA VAD VAK VAY VBG VCT VCV VDF VDZ VEE VEL VGT VIS VLD VNW 
```

```
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
VNY VOK VPC VPS VRB VSF VYS W04 W13 WAA WAL WAS WBB WBQ WBU WBW WDR WFB WFK WHD 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
WHP WIH WKK WKL WLK WMO WRB WRG WRI WRL WSD WSJ WSN WST WSX WTK WTL WWD WWP WWT 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
WYS X01 X04 X07 X21 X26 X39 X49 X59 XFL XNA XZK Y51 Y72 YAK YIP YKM YKN YNG YUM 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
Z84 ZBP ZFV ZPH ZRA ZRD ZRP ZRT ZRZ ZSF ZSY ZTF ZTY ZUN ZVE ZWI ZWU ZYP 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
```

```r
# The missing key is: weather$origin  <->  airports$faa
# Example join to attach airport metadata to each weather row:
weather_with_airport <- weather %>%
  left_join(airports %>% select(faa, name, lat, lon, tz), by = c("origin" = "faa"))
# peek to confirm the relationship
weather_with_airport %>% select(origin, name, time_hour) %>% slice_head(n = 5)
```

```
# A tibble: 5 × 3
  origin name                time_hour          
  <chr>  <chr>               <dttm>             
1 EWR    Newark Liberty Intl 2013-01-01 01:00:00
2 EWR    Newark Liberty Intl 2013-01-01 02:00:00
3 EWR    Newark Liberty Intl 2013-01-01 03:00:00
4 EWR    Newark Liberty Intl 2013-01-01 04:00:00
5 EWR    Newark Liberty Intl 2013-01-01 05:00:00
```

Reports: After reviewing the dataset, the missing relationship between weather and airports datasets was the code of ariline (weather$originandairports$faa). In the weather datasets, the “origin” colomn has only three airports categories (EWR, JFK, and LGA). The coding above was my conducting to merge two datasets based on the weather$originandairports$faa.

Question 5

```r
weather_keyed <- weather %>%
  mutate(
    hw_key = str_c(year, month, day, hour, origin, sep = "-")
  )

dup_count <- sum(duplicated(weather_keyed$hw_key))
dup_breakdown <- weather_keyed %>%
  count(year, month, day, hour, origin, name = "n") %>%
  arrange(desc(n)) %>%
  filter(n > 1)
```

```
dup_count
```

```
[1] 3
```

```
head(dup_breakdown)
```

```
# A tibble: 3 × 6
   year month   day  hour origin     n
  <int> <int> <int> <int> <chr>  <int>
1  2013    11     3     1 EWR        2
2  2013    11     3     1 JFK        2
3  2013    11     3     1 LGA        2
```

Reports: Based on the result above, there are 3 pairs of duplicated values. It might because multiple measurements can be recorded within the same hour at an airport. Therefore, we got >1 row per hour origin.

Merge weather onto each flight by scheduled departure hour & origin

```
flights_weather <- flights %>%
  select(year, month, day, dep_time, sched_dep_time, dep_delay, arr_delay,
         origin, dest, time_hour, flight, carrier, tailnum) %>%
  left_join(weather %>% select(origin, time_hour, temp, dewp, humid, wind_dir,
                               wind_speed, wind_gust, precip, pressure, visib),
            by = c("origin","time_hour"))

dim(flights_weather); flights_weather %>% glimpse()
```

```
[1] 336776     22
```

```
Rows: 336,776
Columns: 22
$ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
$ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, …
$ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, …
$ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1…
$ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1…
$ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",…
$ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",…
$ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
$ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4…
$ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "…
$ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394…
$ temp           <dbl> 39.02, 39.92, 39.02, 39.02, 39.92, 39.02, 37.94, 39.92,…
$ dewp           <dbl> 28.04, 24.98, 26.96, 26.96, 24.98, 28.04, 28.04, 24.98,…
$ humid          <dbl> 64.43, 54.81, 61.63, 61.63, 54.81, 64.43, 67.21, 54.81,…
$ wind_dir       <dbl> 260, 250, 260, 260, 260, 260, 240, 260, 260, 260, 260, …
$ wind_speed     <dbl> 12.65858, 14.96014, 14.96014, 14.96014, 16.11092, 12.65…
$ wind_gust      <dbl> NA, 21.86482, NA, NA, 23.01560, NA, NA, 23.01560, NA, 2…
$ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ pressure       <dbl> 1011.9, 1011.4, 1012.1, 1012.1, 1011.7, 1011.9, 1012.4,…
$ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
```

```
# Each flight now carries the *departure-hour* weather at its origin.
```

Question 6

For this question, I referred the checklist from the lecture slide in the 3rd week (EDA Checklist: The goal of EDA is to better understand your data. Let’s use the checklist:

2. Check the size of the data

3. Examine the variables and their types

4. Look at the top and bottom of the data

5. Visualize the distributions of key variables

```
# 6a) Size of the data
nrow(flights_weather)      # number of rows (flights)
```

```
[1] 336776
```

```
ncol(flights_weather)      # number of columns (variables)
```

```
[1] 22
```

```
# 6b) Examine variables and their types
glimpse(flights_weather)   # compact structure: names, types, and example values
```

```
Rows: 336,776
Columns: 22
```

```
$ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
$ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, …
$ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, …
$ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1…
$ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1…
$ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",…
$ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",…
$ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
$ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4…
$ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "…
$ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394…
$ temp           <dbl> 39.02, 39.92, 39.02, 39.02, 39.92, 39.02, 37.94, 39.92,…
$ dewp           <dbl> 28.04, 24.98, 26.96, 26.96, 24.98, 28.04, 28.04, 24.98,…
$ humid          <dbl> 64.43, 54.81, 61.63, 61.63, 54.81, 64.43, 67.21, 54.81,…
$ wind_dir       <dbl> 260, 250, 260, 260, 260, 260, 240, 260, 260, 260, 260, …
$ wind_speed     <dbl> 12.65858, 14.96014, 14.96014, 14.96014, 16.11092, 12.65…
$ wind_gust      <dbl> NA, 21.86482, NA, NA, 23.01560, NA, NA, 23.01560, NA, 2…
$ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ pressure       <dbl> 1011.9, 1011.4, 1012.1, 1012.1, 1011.7, 1011.9, 1012.4,…
$ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
```

```
# 6c) Look at the top and bottom of the data
head(flights_weather, 5)   # first 5 rows
```

```
# A tibble: 5 × 22
   year month   day dep_time sched_dep_time dep_delay arr_delay origin dest 
  <int> <int> <int>    <int>          <int>     <dbl>     <dbl> <chr>  <chr>
1  2013     1     1      517            515         2        11 EWR    IAH  
2  2013     1     1      533            529         4        20 LGA    IAH  
3  2013     1     1      542            540         2        33 JFK    MIA  
4  2013     1     1      544            545        -1       -18 JFK    BQN  
5  2013     1     1      554            600        -6       -25 LGA    ATL  
# ℹ 13 more variables: time_hour <dttm>, flight <int>, carrier <chr>,
#   tailnum <chr>, temp <dbl>, dewp <dbl>, humid <dbl>, wind_dir <dbl>,
#   wind_speed <dbl>, wind_gust <dbl>, precip <dbl>, pressure <dbl>,
#   visib <dbl>
```

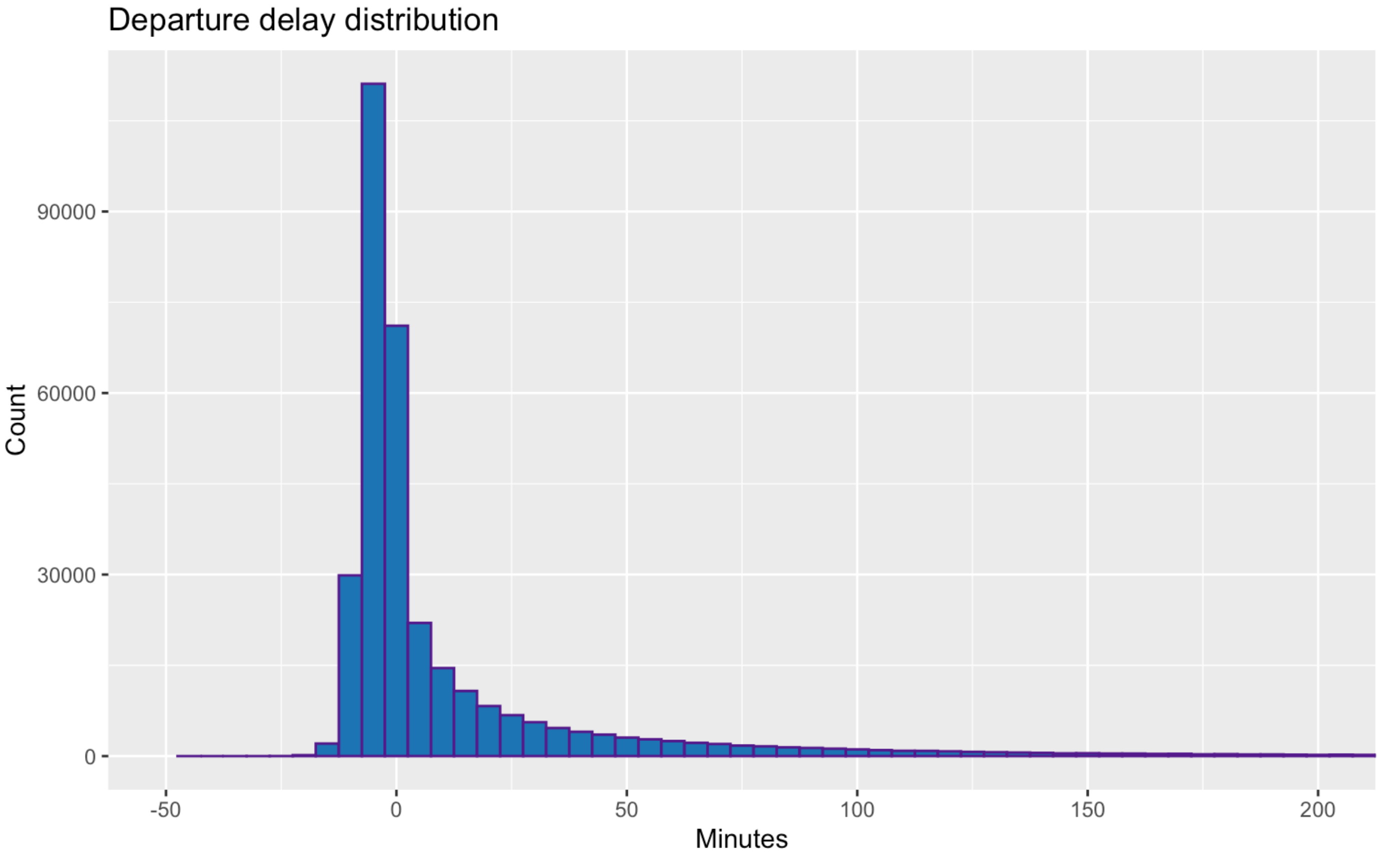
```
tail(flights_weather, 5)   # last 5 rows
```

```
# A tibble: 5 × 22
   year month   day dep_time sched_dep_time dep_delay arr_delay origin dest 
  <int> <int> <int>    <int>          <int>     <dbl>     <dbl> <chr>  <chr>
1  2013     9    30       NA           1455        NA        NA JFK    DCA  
2  2013     9    30       NA           2200        NA        NA LGA    SYR  
3  2013     9    30       NA           1210        NA        NA LGA    BNA  
4  2013     9    30       NA           1159        NA        NA LGA    CLE  
5  2013     9    30       NA            840        NA        NA LGA    RDU  
```

```
# ℹ 13 more variables: time_hour <dttm>, flight <int>, carrier <chr>,
#   tailnum <chr>, temp <dbl>, dewp <dbl>, humid <dbl>, wind_dir <dbl>,
#   wind_speed <dbl>, wind_gust <dbl>, precip <dbl>, pressure <dbl>,
#   visib <dbl>
```

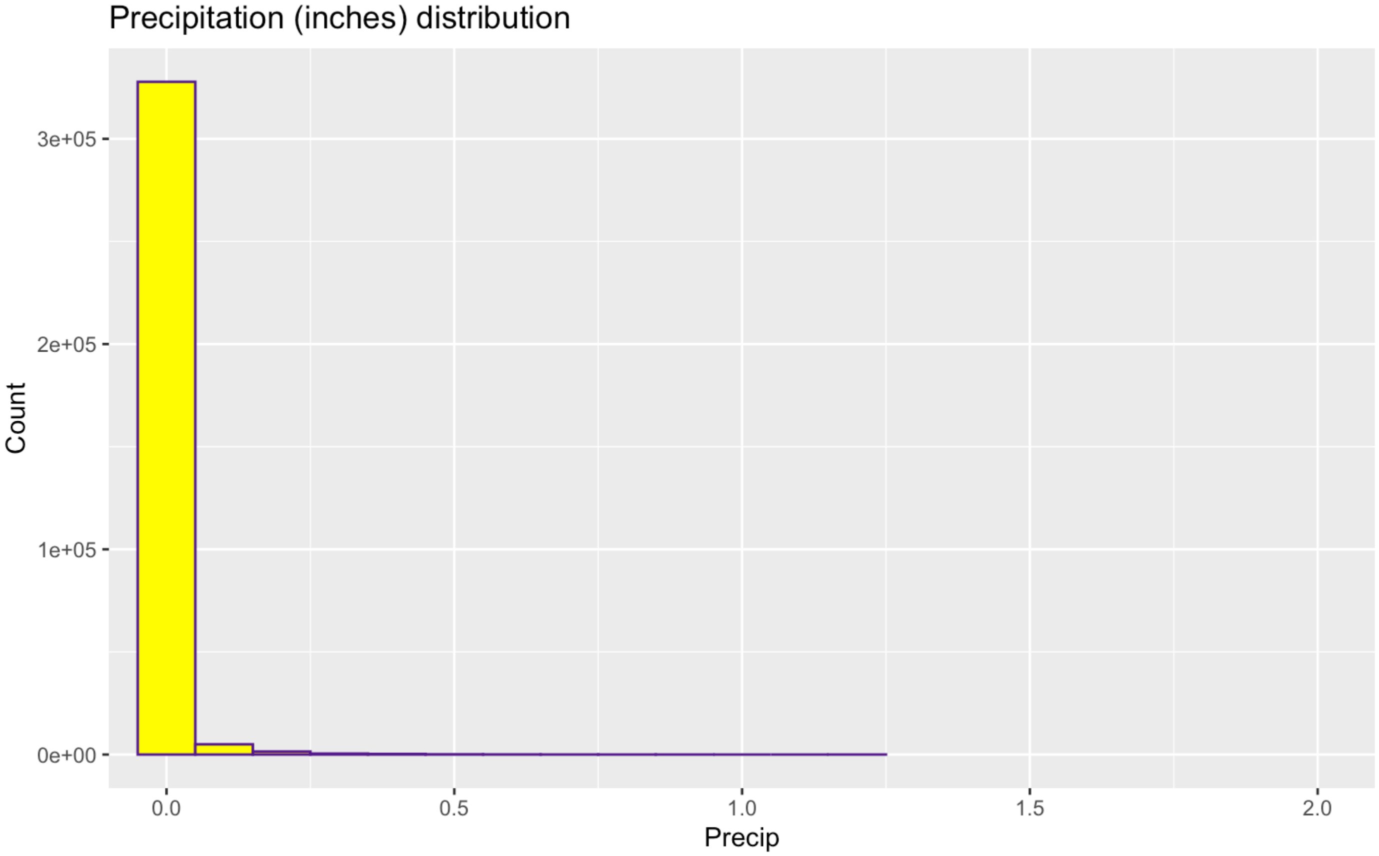
```
# 6d) Visualize distributions of key variables related to delays & weather
# Departure delay distribution (trim extreme to visualize; delays are in minutes)
ggplot(flights_weather, aes(x = dep_delay)) +             # histogram of departure
  geom_histogram(binwidth = 5, fill = "#1f78b4", color = "purple4") +
  coord_cartesian(xlim = c(-50, 200)) +                   # focus on common range
  labs(title = "Departure delay distribution", x = "Minutes", y = "Count")
```

```
Warning: Removed 8255 rows containing non-finite outside the scale range
(`stat_bin()`).
```



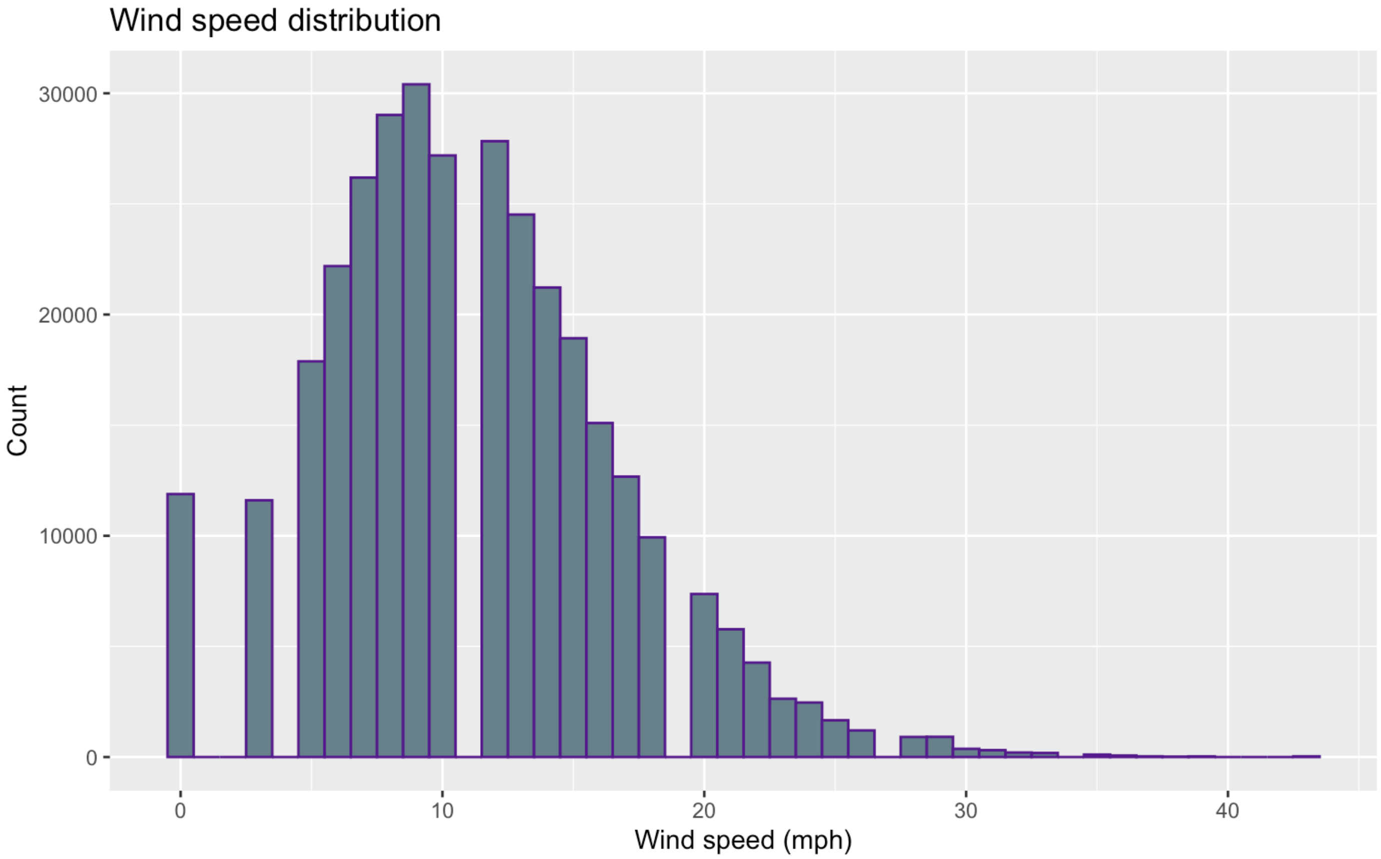
```
# Weather: precipitation (many zeros, heavy right tail)
ggplot(flights_weather, aes(x = precip)) +
  geom_histogram(binwidth = 0.1, fill = "yellow1", color = "purple4") +
  coord_cartesian(xlim = c(0, 2)) +
  labs(title = "Precipitation (inches) distribution", x = "Precip", y = "Count")
```

```
Warning: Removed 1556 rows containing non-finite outside the scale range
(`stat_bin()`).
```



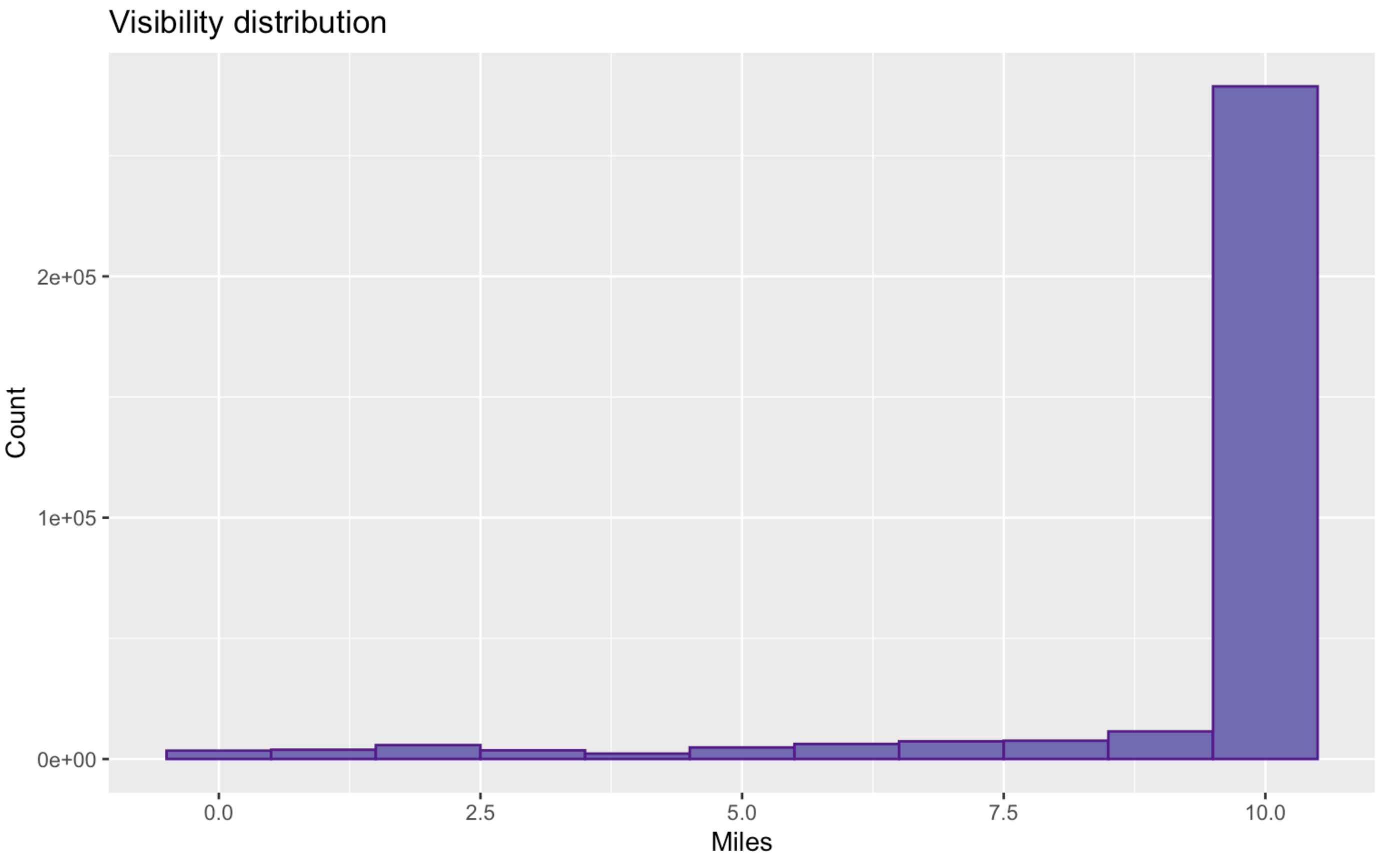
```
# Weather: wind speed
ggplot(flights_weather, aes(x = wind_speed)) +
  geom_histogram(binwidth = 1,fill = "lightblue4", color = "purple4") +
  labs(title = "Wind speed distribution", x = "Wind speed (mph)", y = "Count")
```

```
Warning: Removed 1634 rows containing non-finite outside the scale range
(`stat_bin()`).
```



```
# Weather: visibility
ggplot(flights_weather, aes(x = visib)) +
  geom_histogram(binwidth = 1, fill = "#7570b3", color = "purple4") +
  labs(title = "Visibility distribution", x = "Miles", y = "Count")
```

```
Warning: Removed 1556 rows containing non-finite outside the scale range
(`stat_bin()`).
```

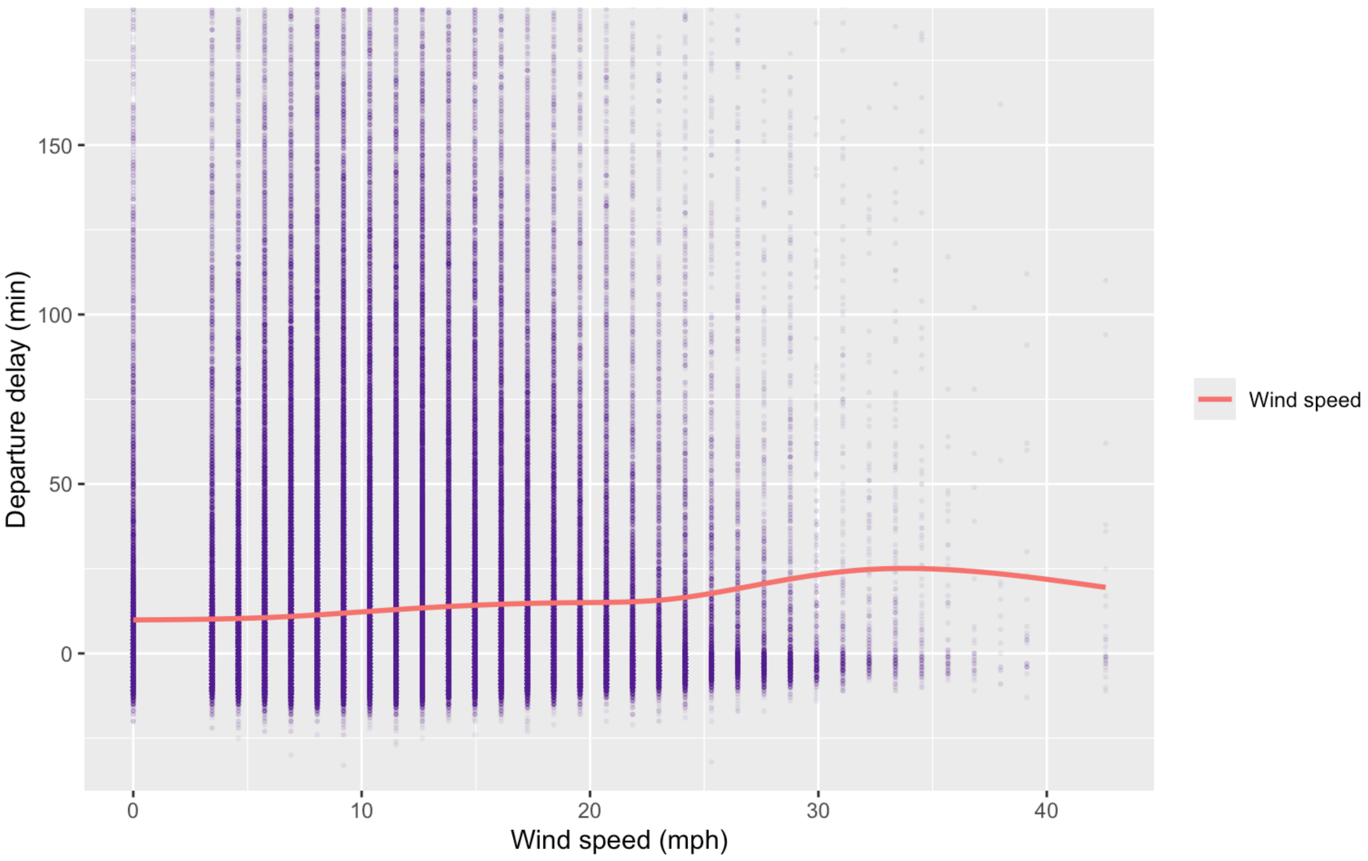


```r
# Quick expectation check: scatter of dep_delay vs key weather features
# Each plot maps a constant label to 'color' so a legend exists to collect
p_wind <- ggplot(flights_weather, aes(wind_speed, dep_delay, color = "Wind speed")) +
  geom_point(alpha = 0.05, size = 0.4, color = "purple4", na.rm = TRUE) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), se = FALSE, size = 1) +
  coord_cartesian(ylim = c(-30, 180)) +
  labs(x = "Wind speed (mph)", y = "Departure delay (min)", color = NULL)
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```

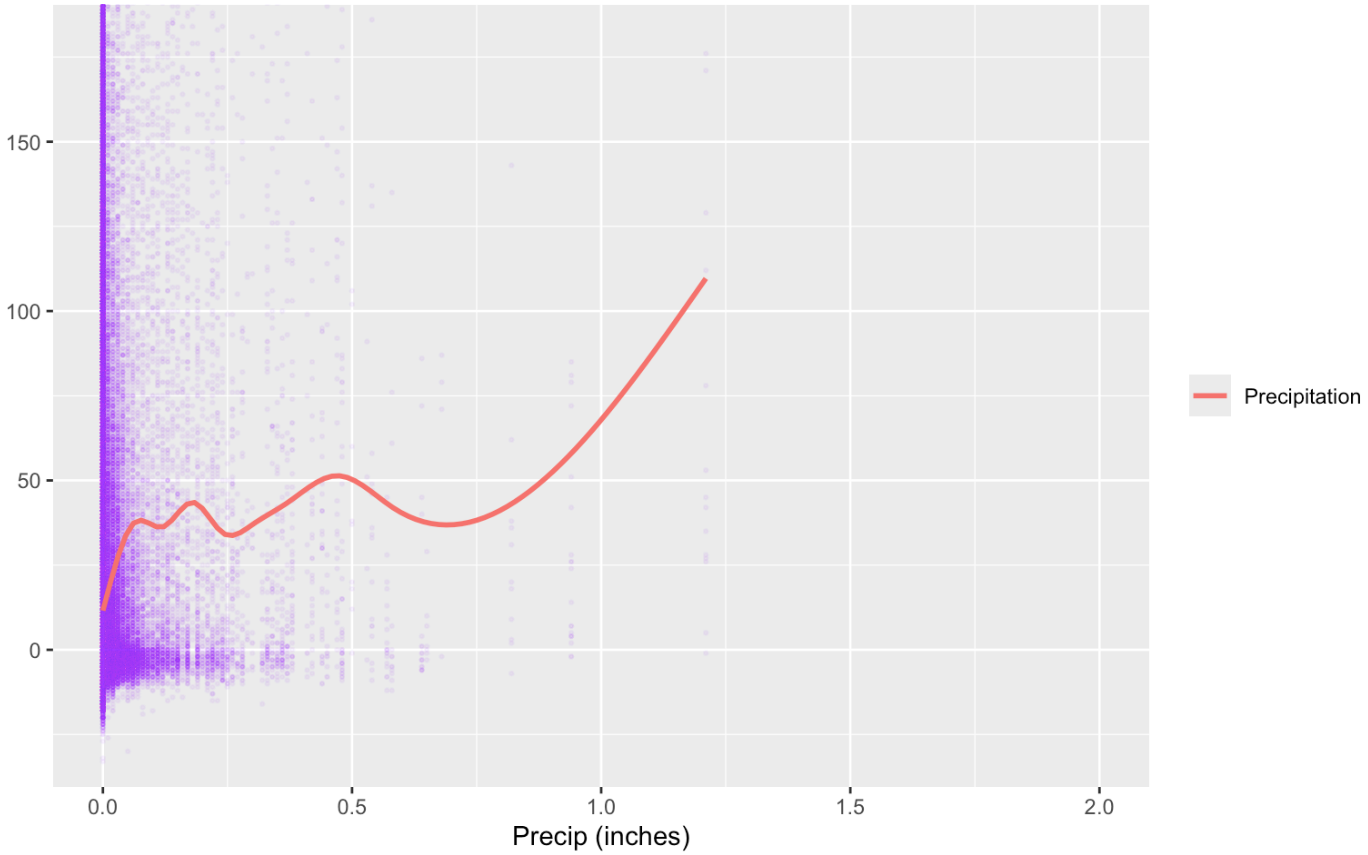
```r
p_prec <- ggplot(flights_weather, aes(precip, dep_delay, color = "Precipitation")) +
  geom_point(alpha = 0.05, size = 0.4, color = "purple1", na.rm = TRUE) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), se = FALSE, size = 1) +
  coord_cartesian(xlim = c(0, 2), ylim = c(-30, 180)) +
  labs(x = "Precip (inches)", y = NULL, color = NULL)

p_vis  <- ggplot(flights_weather, aes(visib, dep_delay, color = "Visibility")) +
  geom_point(alpha = 0.05, size = 0.4, color = "blue", na.rm = TRUE) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), se = FALSE, size = 1) +
  coord_cartesian(ylim = c(-30, 180)) +
  labs(x = "Visibility (miles)", y = NULL, color = NULL)

p_wind
```

```
Warning: Removed 9861 rows containing non-finite outside the scale range
(`stat_smooth()`).
```



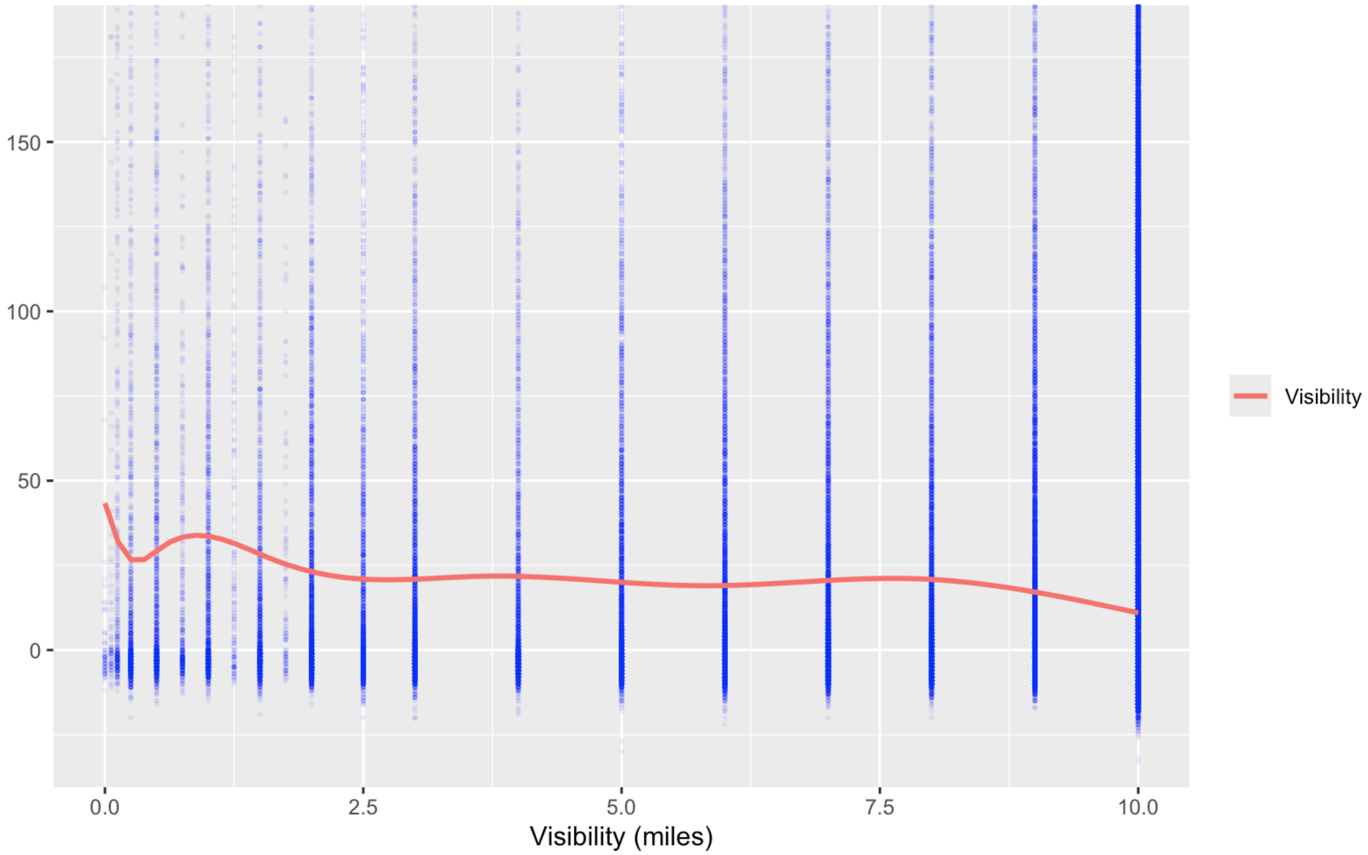
```
p_prec
```

```
Warning: Removed 9783 rows containing non-finite outside the scale range
(`stat_smooth()`).
```



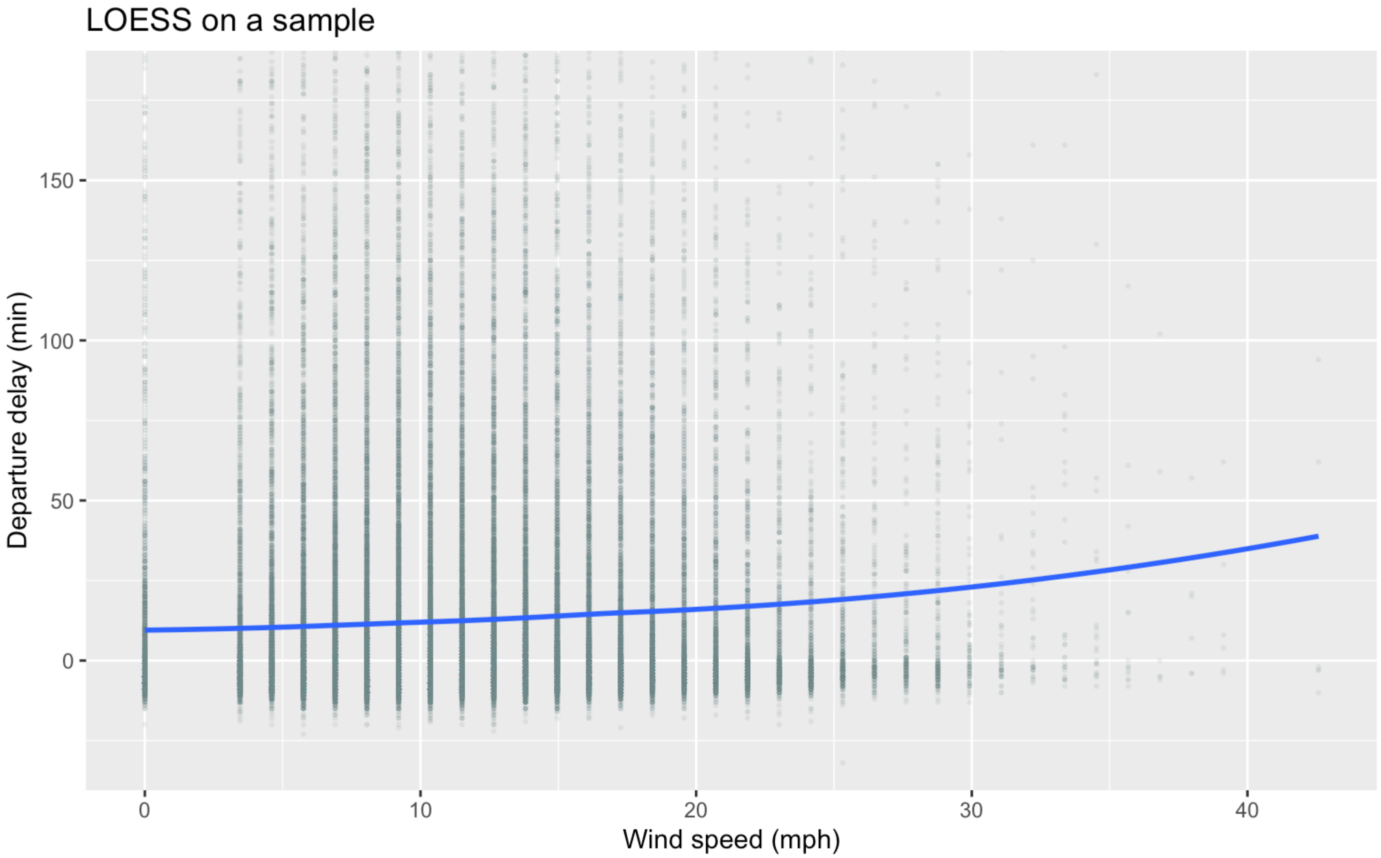
```
p_vis
```

```
Warning: Removed 9783 rows containing non-finite outside the scale range
(`stat_smooth()`).
```



```r
ggplot(
  flights_weather %>%
    filter(is.finite(dep_delay), is.finite(wind_speed)) %>%       # drop NAs first
    slice_sample(n = 80000),                                      # smaller sample
  aes(wind_speed, dep_delay)
) +
  geom_point(alpha = 0.08, size = 0.4, color = "lightblue4") +
  geom_smooth(method = "loess", se = FALSE, span = 0.8) +         # LOESS on a sample is
  coord_cartesian(ylim = c(-30, 180)) +
  labs(title = "LOESS on a sample", x = "Wind speed (mph)", y = "Departure delay (min)")
```

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



The result above are my performing follwed steps 2-5 of the EDA checklist presented in class.

Question 7

```
# Helper to keep only flights with a reported dep_delay
fw <- flights_weather %>% filter(!is.na(dep_delay))

# 7a. Average departure delay by *day*
daily <- fw %>%
  group_by(year, month, day) %>%
  summarise(avg_dep_delay = mean(dep_delay), n = n(), .groups = "drop") %>%
  arrange(desc(avg_dep_delay))
head(daily, 1)    # worst day
```

```
# A tibble: 1 × 5
   year month   day avg_dep_delay     n
  <int> <int> <int>         <dbl> <int>
1  2013     3     8          83.5   799
```

```
# 7b. By day × origin
daily_org <- fw %>%
```

```r
  group_by(origin, year, month, day) %>%
  summarise(avg_dep_delay = mean(dep_delay), n = n(), .groups = "drop") %>%
  arrange(desc(avg_dep_delay))
head(daily_org, 1) # worst airport-day
```

```
# A tibble: 1 × 6
  origin  year month   day avg_dep_delay     n
  <chr>  <int> <int> <int>         <dbl> <int>
1 LGA     2013     3     8          106.   229
```

```r
# 7c. By hour × origin
hourly_org <- fw %>%
  mutate(hour = hour(time_hour)) %>%
  group_by(origin, year, month, day, hour) %>%
  summarise(avg_dep_delay = mean(dep_delay), n = n(), .groups = "drop") %>%
  arrange(desc(avg_dep_delay))
head(hourly_org, 1) # worst airport-hour
```

```
# A tibble: 1 × 7
  origin  year month   day  hour avg_dep_delay     n
  <chr>  <int> <int> <int> <int>         <dbl> <int>
1 LGA     2013     7    28    21          280.     3
```

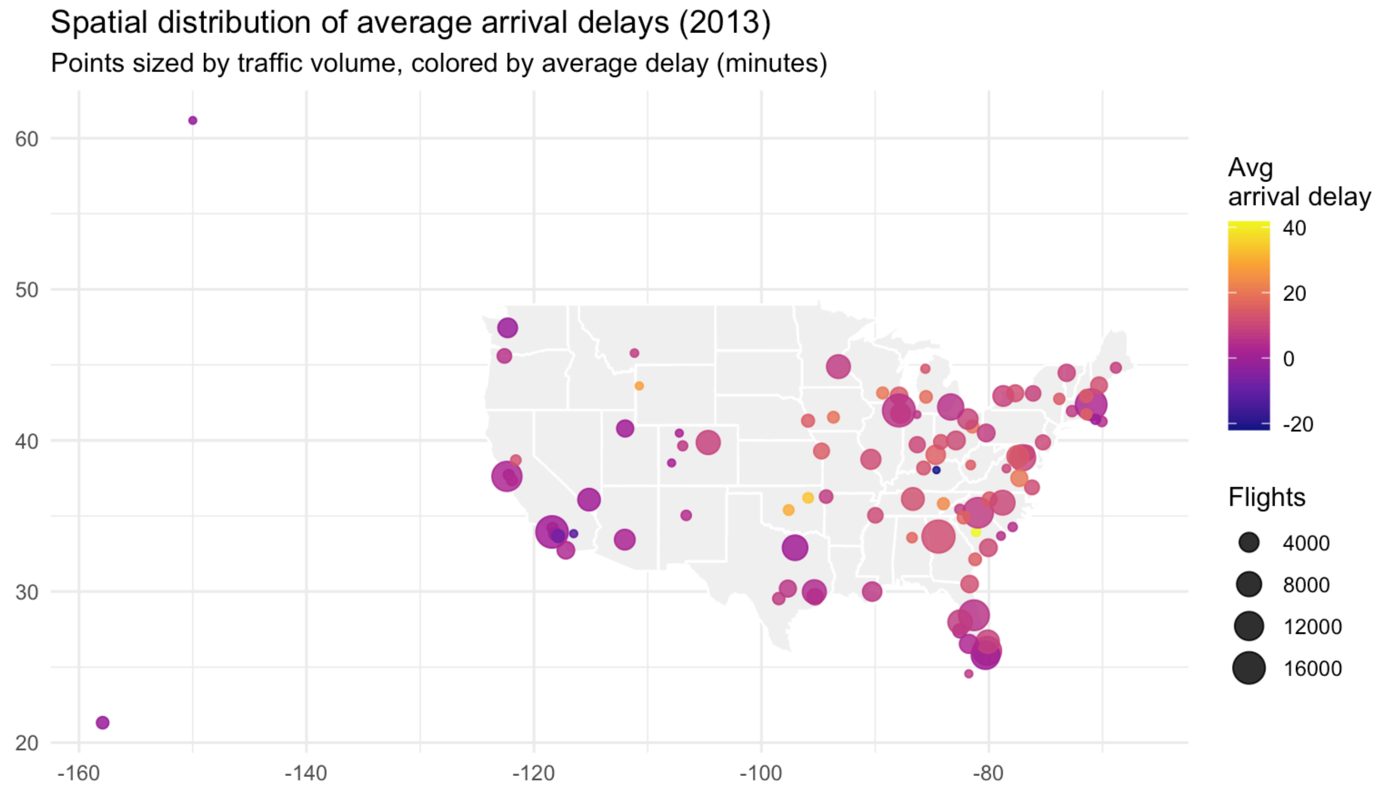
Base on the result above, when grouping by day and day along with origin, the worst average departure delay occurred on March 8th with an average delay of 83.53692 minutes and 105.7249 minutes, repectively. The letter occurred on LGA. When grouping by hour and origin, the worst average departure delay occurred on July 28th at LGA at 9 PM with an average delay of 279.6667 minutes.

Question 8

```r
# Average arrival delay by destination airport (dest)
dest_avgs <- flights %>%
  filter(!is.na(arr_delay)) %>%
  group_by(dest) %>%
  summarise(avg_arr_delay = mean(arr_delay), n = n(), .groups = "drop")

airports_delay <- airports %>%
  inner_join(dest_avgs, by = c("faa" = "dest"))

usa <- map_data("state")
summary(airports_delay$avg_arr_delay)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-22.000   4.548   8.369   8.934  12.672  41.764 
```

```
# map
ggplot() +
  geom_polygon(data = usa, aes(long, lat, group = group),
               fill = "grey95", color = "white") +
  geom_point(data = airports_delay,
             aes(lon, lat, color = avg_arr_delay, size = n),
             alpha = 0.85) +
  scale_color_viridis_c(option = "plasma", name = "Avg\narrival delay") +
  scale_size_continuous(range = c(1, 6), name = "Flights") +
  coord_quickmap() +
  labs(title = "Spatial distribution of average arrival delays (2013)",
       subtitle = "Points sized by traffic volume, colored by average delay (minutes)",
       x = NULL, y = NULL) +
  theme_minimal()
```


Spatial distribution of average arrival delays (2013)
Points sized by traffic volume, colored by average delay (minutes)
60
50
40
30
20
-160
-140
-120
-100
-80
Avg
arrival delay
40
20
0
-20
Flights
4000
8000
12000
16000

Question 9

```
# 9a) Create binned weather categories to summarize relationships cleanly
merged_binned <- flights_weather %>%
  mutate(
    precip_bin = cut(precip, breaks = c(-Inf, 0, 0.1, 0.5, 1, Inf),  # none, light, mod,
                     labels = c("0", "(0,0.1]", "(0.1,0.5]", "(0.5,1]", ">1")),
    wind_bin   = cut(wind_speed, breaks = c(-Inf, 5, 10, 20, Inf),   # calm, light, breez
                     labels = c("≤5", "(5,10]", "(10,20]", ">20")),
    visib_bin  = cut(visib, breaks = c(-Inf, 2, 5, 10, Inf),         # poor, fair, good,
                     labels = c("≤2", "(2,5]", "(5,10]", ">10"))
  )
```

```
# 9b) Summaries: mean departure delay by each phenomenon
sum_precip <- merged_binned %>%
  group_by(precip_bin) %>%
  summarise(mean_dep_delay = mean(dep_delay, na.rm = TRUE), n = n()) %>%
  arrange(desc(mean_dep_delay))

sum_wind <- merged_binned %>%
  group_by(wind_bin) %>%
  summarise(mean_dep_delay = mean(dep_delay, na.rm = TRUE), n = n()) %>%
  arrange(desc(mean_dep_delay))

sum_visib <- merged_binned %>%
  group_by(visib_bin) %>%
  summarise(mean_dep_delay = mean(dep_delay, na.rm = TRUE), n = n()) %>%
  arrange(mean_dep_delay)                                       # lower visibility → us

sum_precip; sum_wind; sum_visib                                # print summaries
```

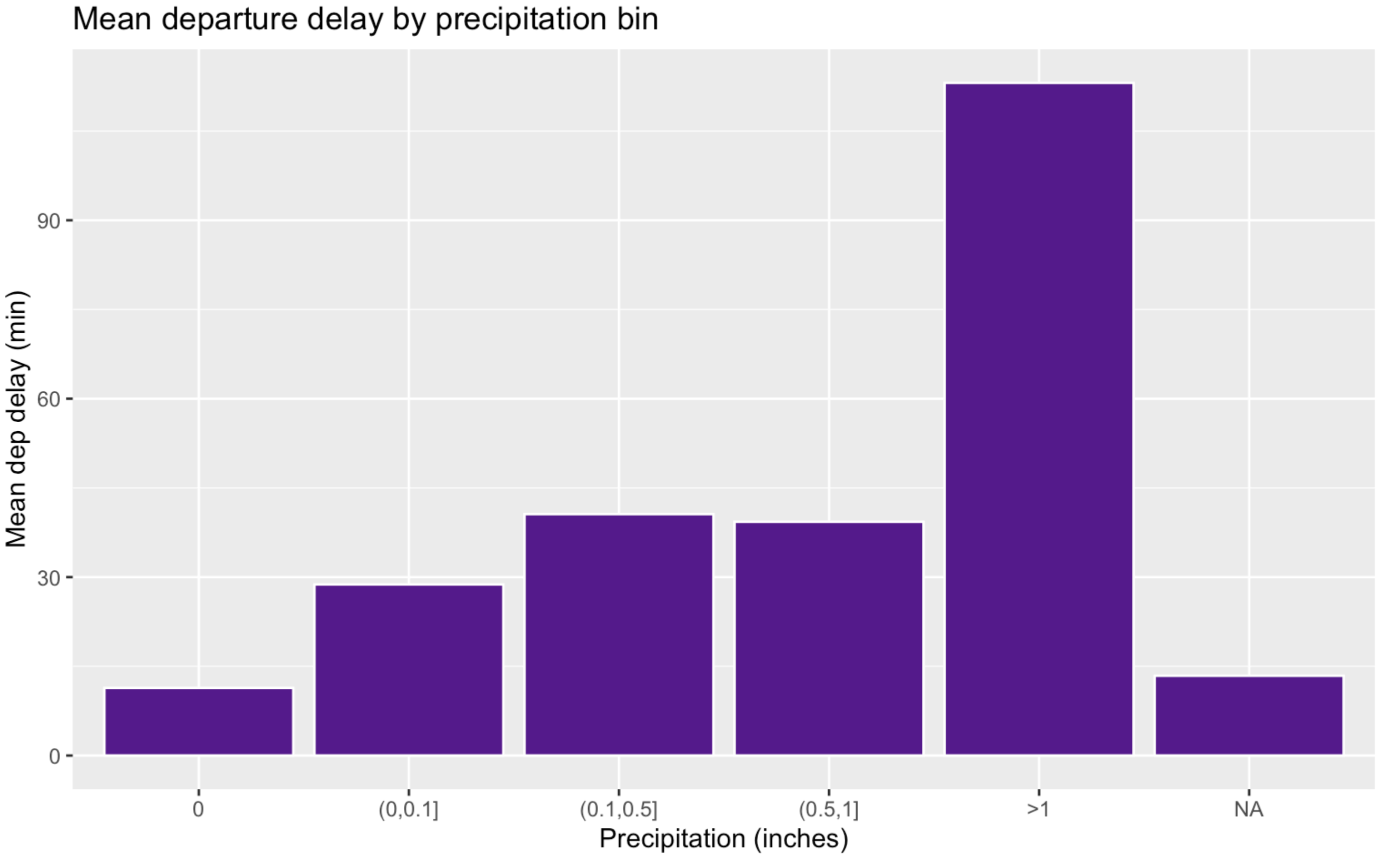
```
# A tibble: 6 × 3
  precip_bin mean_dep_delay      n
  <fct>               <dbl>  <int>
1 >1                  113.      21
2 (0.1,0.5]            40.6   3914
3 (0.5,1]              39.3    154
4 (0,0.1]              28.8  18913
5 <NA>                 13.4   1556
6 0                    11.4 312218
```

```
# A tibble: 5 × 3
  wind_bin mean_dep_delay      n
  <fct>             <dbl>  <int>
1 >20                16.9  21194
2 (10,20]            13.6 164770
3 <NA>               13.0   1634
4 (5,10]             11.3 107806
5 ≤5                 10.2  41372
```

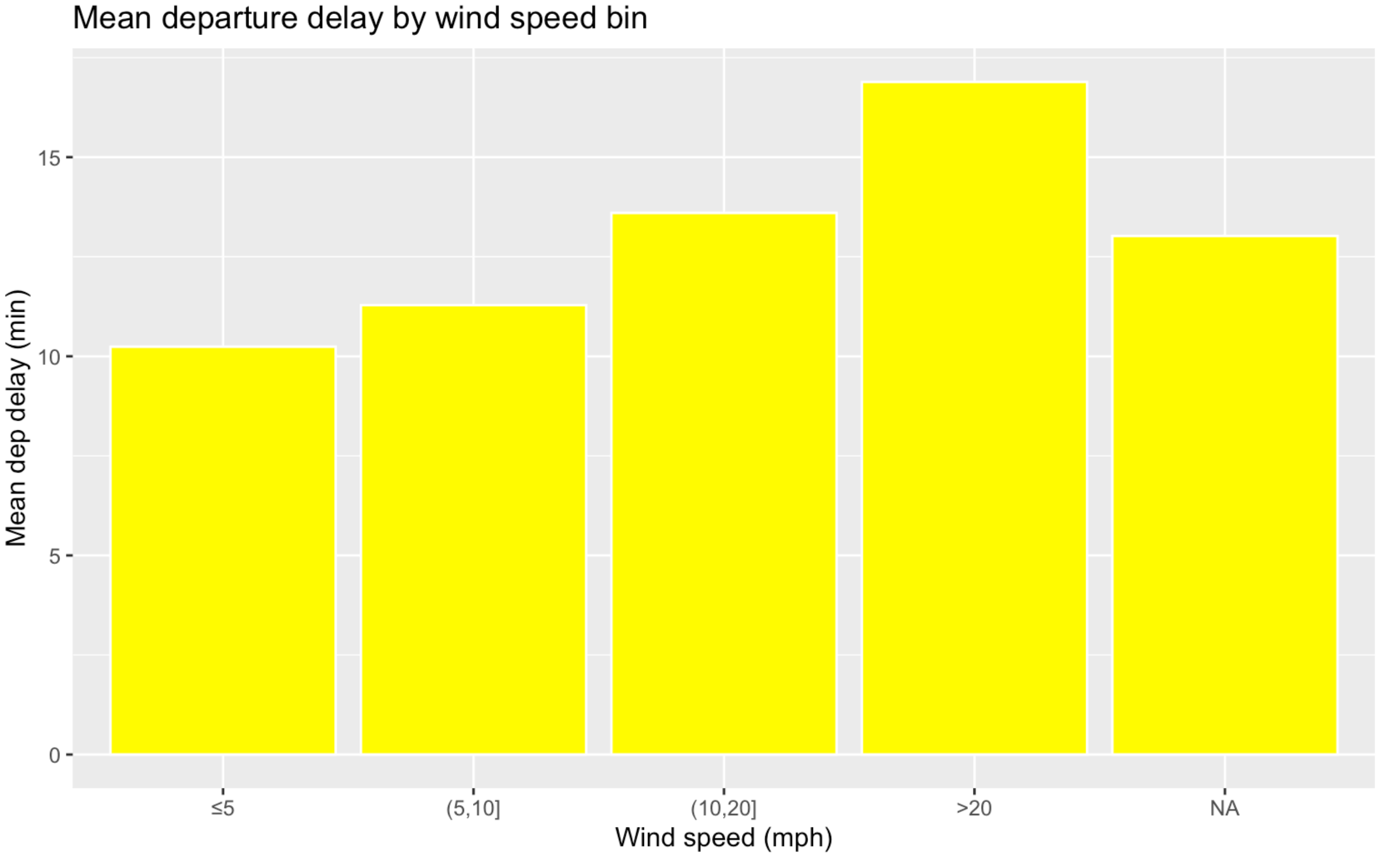
```
# A tibble: 4 × 3
  visib_bin mean_dep_delay      n
  <fct>              <dbl>  <int>
1 (5,10]              11.8 311396
2 <NA>                13.4   1556
```

```
3 (2,5]              20.9  13120
4 ≤2                 28.0  10704
```

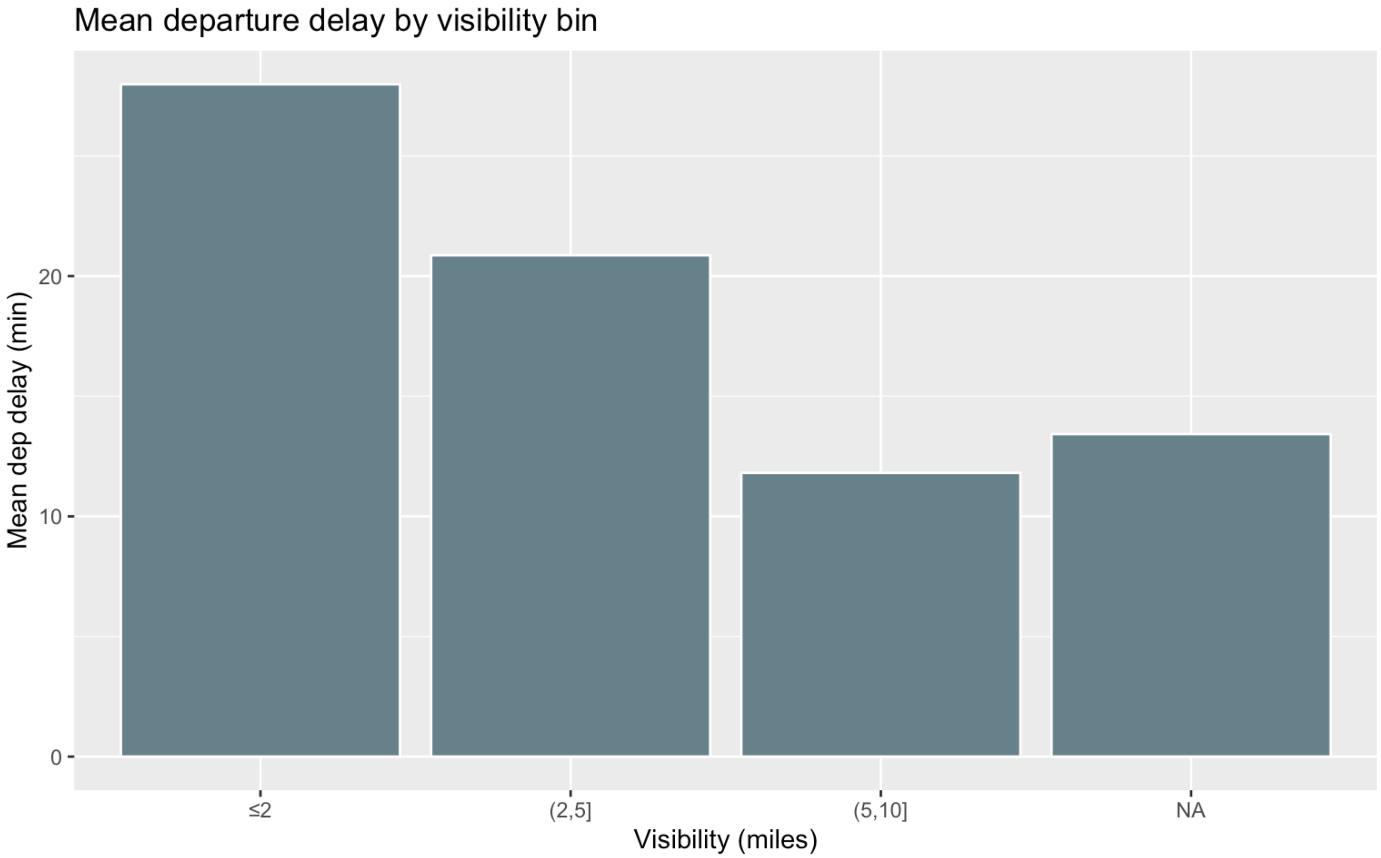
```
# 9c) Visualization: which look worst?
ggplot(sum_precip, aes(precip_bin, mean_dep_delay)) +
  geom_col(fill = "purple4", color = "white") +  # select color
  labs(title = "Mean departure delay by precipitation bin",
       x = "Precipitation (inches)", y = "Mean dep delay (min)")
```



```
ggplot(sum_wind, aes(wind_bin, mean_dep_delay)) +
  geom_col(fill = "yellow", color = "white") +
  labs(title = "Mean departure delay by wind speed bin",
       x = "Wind speed (mph)", y = "Mean dep delay (min)")
```



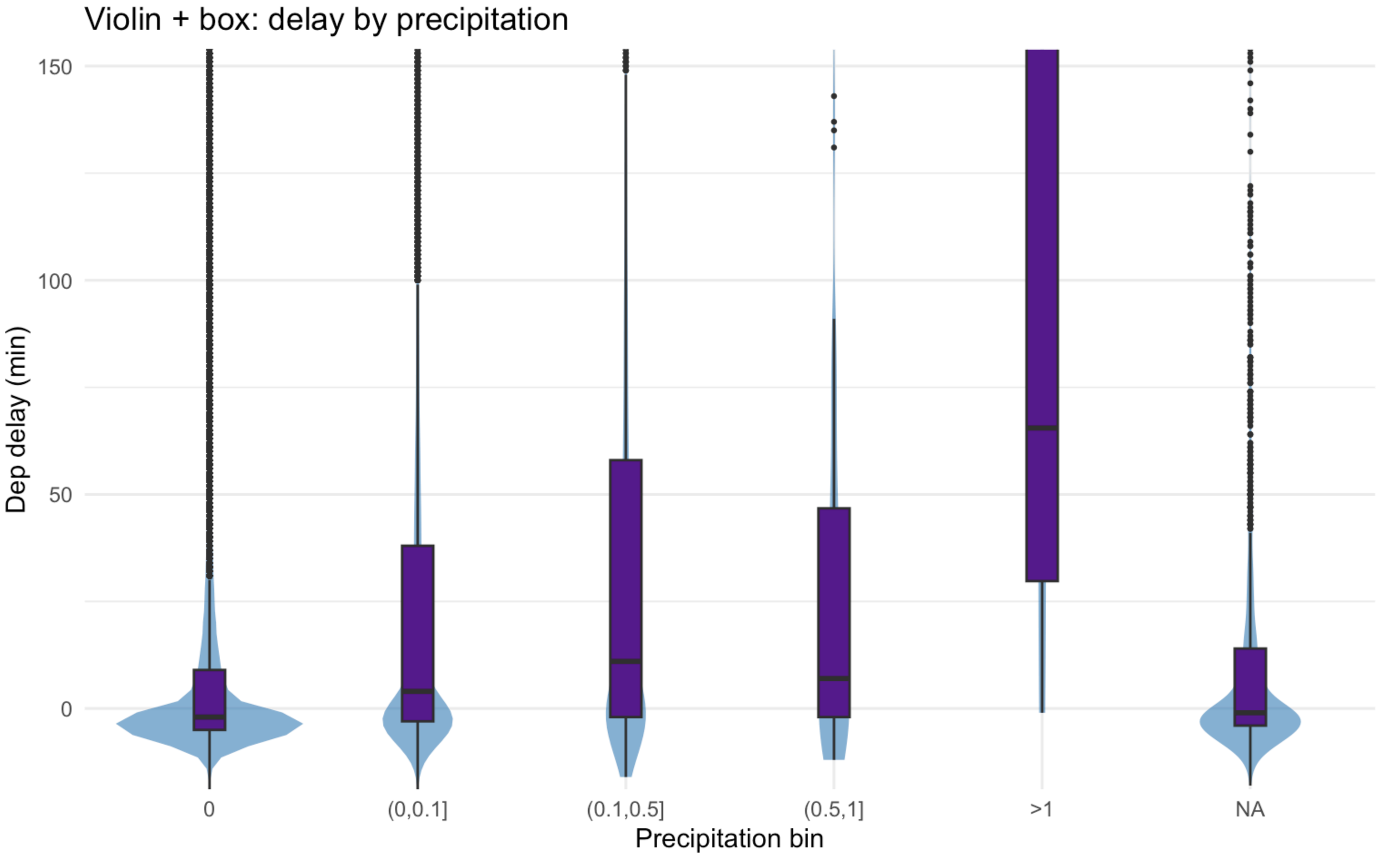
```r
ggplot(sum_visib, aes(visib_bin, mean_dep_delay)) +
  geom_col(fill = "lightblue4", color = "white") +
  labs(title = "Mean departure delay by visibility bin",
       x = "Visibility (miles)", y = "Mean dep delay (min)")
```



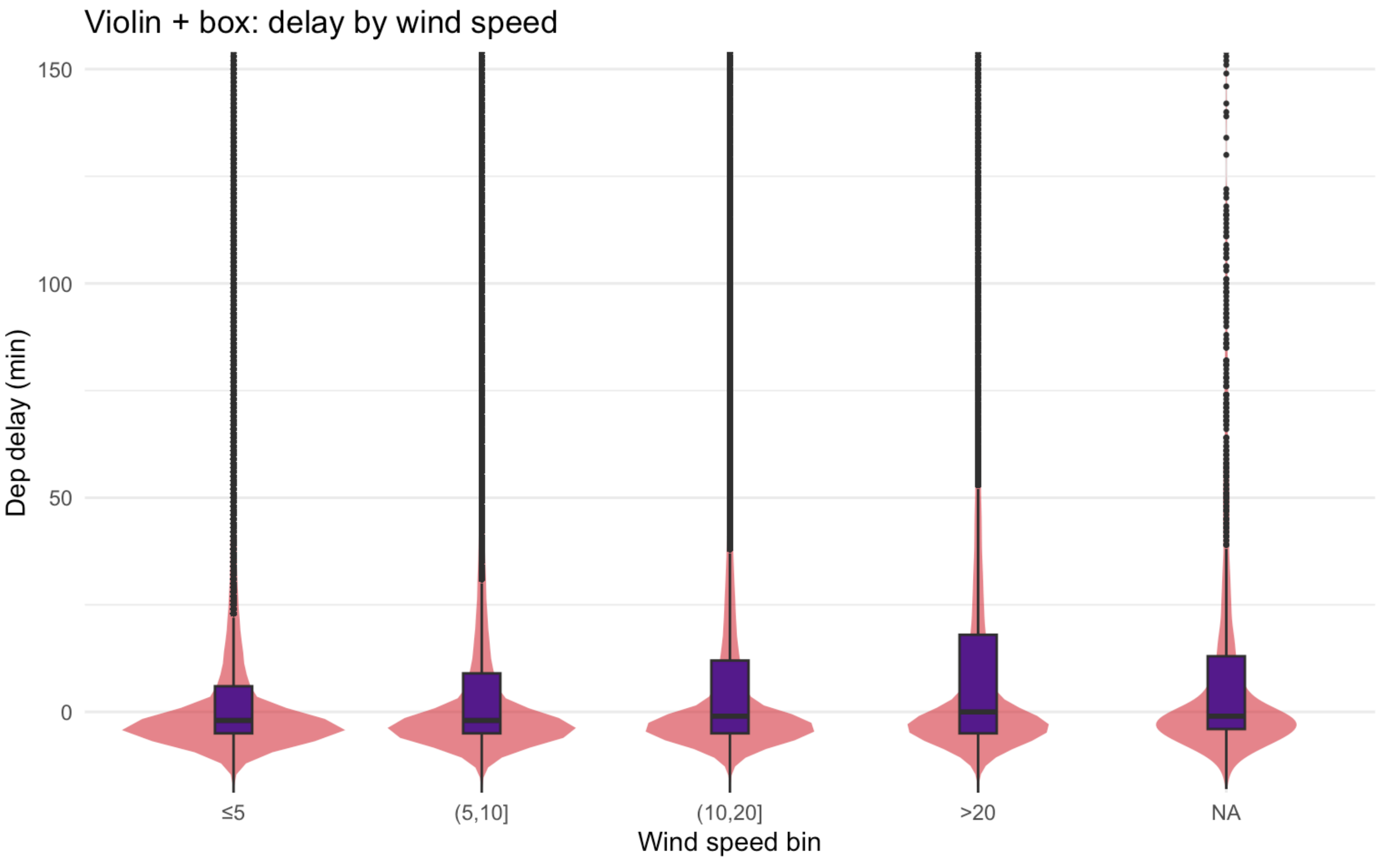
```
# Edit colors
col_precip  <- "#1f77b4"      # precipitation color
col_wind    <- "#d62728"      # wind color
col_visib   <- "#2ca02c"      # visibility color
col_points  <- "grey35"       # point cloud color
col_smooth  <- "#9467bd"      # smooth line color
col_bins    <- c("lightblue1", "#6baed6", "#08306b")          # low→mid→high for heatmaps
col_corr    <- c("#b2182b", "#f7f7f7", "#2166ac")          # neg→0→pos for corr heatmap
```

```
# Bin the weather variables once
fw_binned <- flights_weather %>%
  mutate(
    precip_bin = cut(precip, breaks = c(-Inf, 0, 0.1, 0.5, 1, Inf),
                     labels = c("0", "(0,0.1]", "(0.1,0.5]", "(0.5,1]", ">1")),
    wind_bin   = cut(wind_speed, breaks = c(-Inf, 5, 10, 20, Inf),
                     labels = c("≤5", "(5,10]", "(10,20]", ">20")),
    visib_bin  = cut(visib, breaks = c(-Inf, 2, 5, 10, Inf),
                     labels = c("≤2", "(2,5]", "(5,10]", ">10"))
  )
```

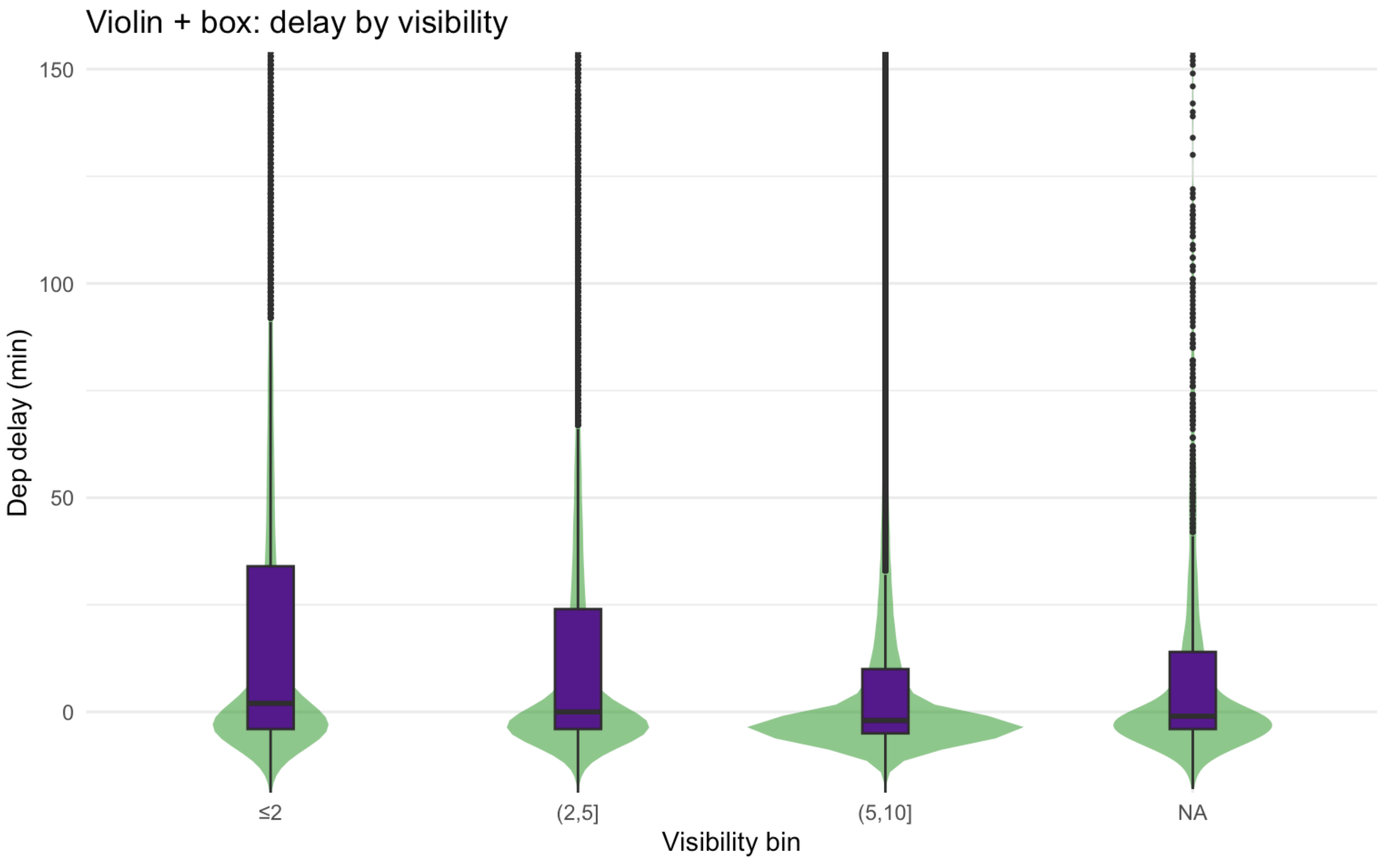
```
# helpful trimmed view (reduce long tail to make box/violin readable)
q_lim <- quantile(flights_weather$dep_delay, c(.02, .98), na.rm = TRUE)
ggplot(fw_binned, aes(precip_bin, dep_delay)) +
  geom_violin(fill = scales::alpha(col_precip, .6), color = NA, na.rm = TRUE) +
  geom_boxplot(width = .15, outlier.size = .5, fill = "purple4", na.rm = TRUE) +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Violin + box: delay by precipitation", x = "Precipitation bin", y = "Dep
  theme_minimal()
```



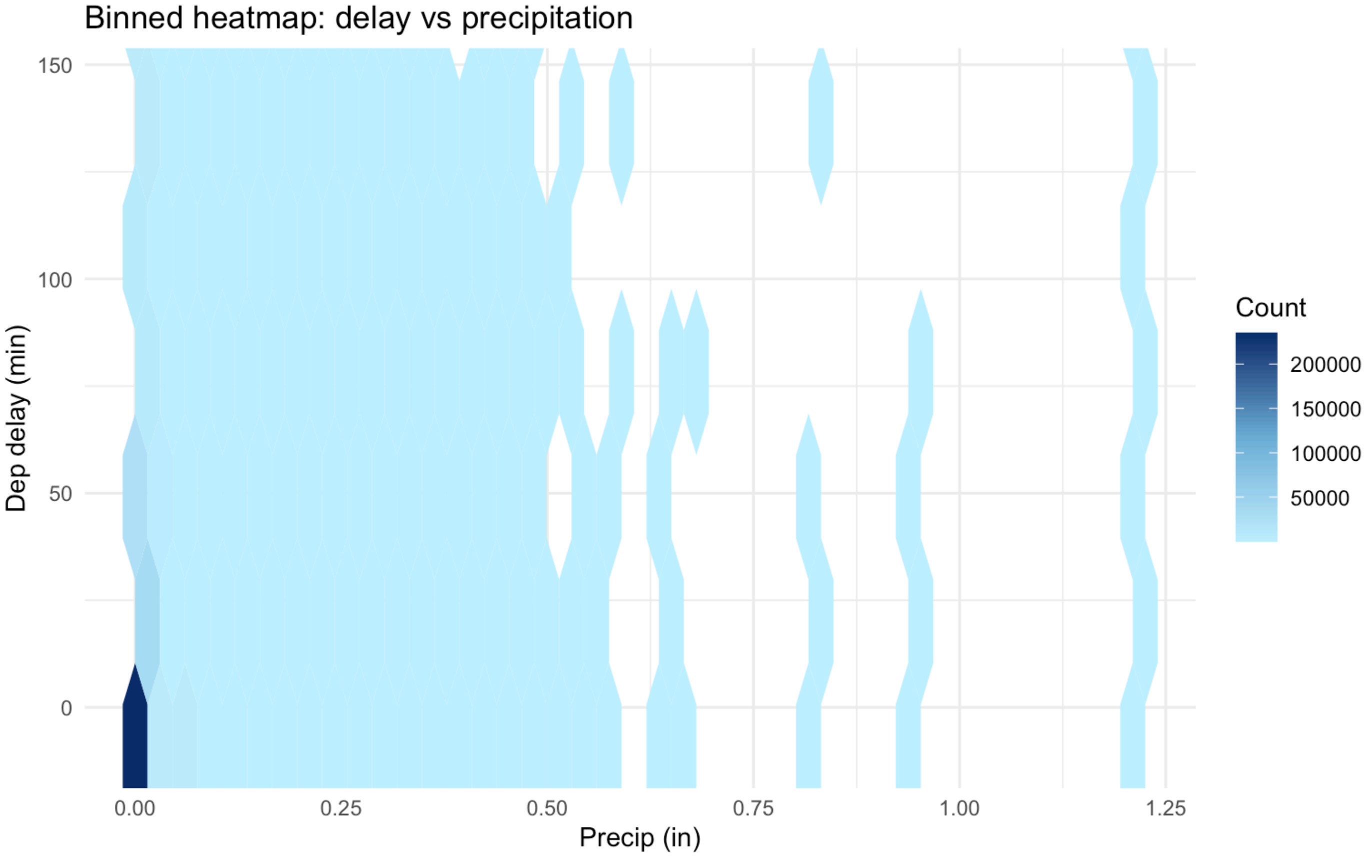
```
ggplot(fw_binned, aes(wind_bin, dep_delay)) +
  geom_violin(fill = scales::alpha(col_wind, .6), color = NA, na.rm = TRUE) +
  geom_boxplot(width = .15, outlier.size = .5, fill ="purple4", na.rm = TRUE) +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Violin + box: delay by wind speed", x = "Wind speed bin", y = "Dep delay
  theme_minimal()
```



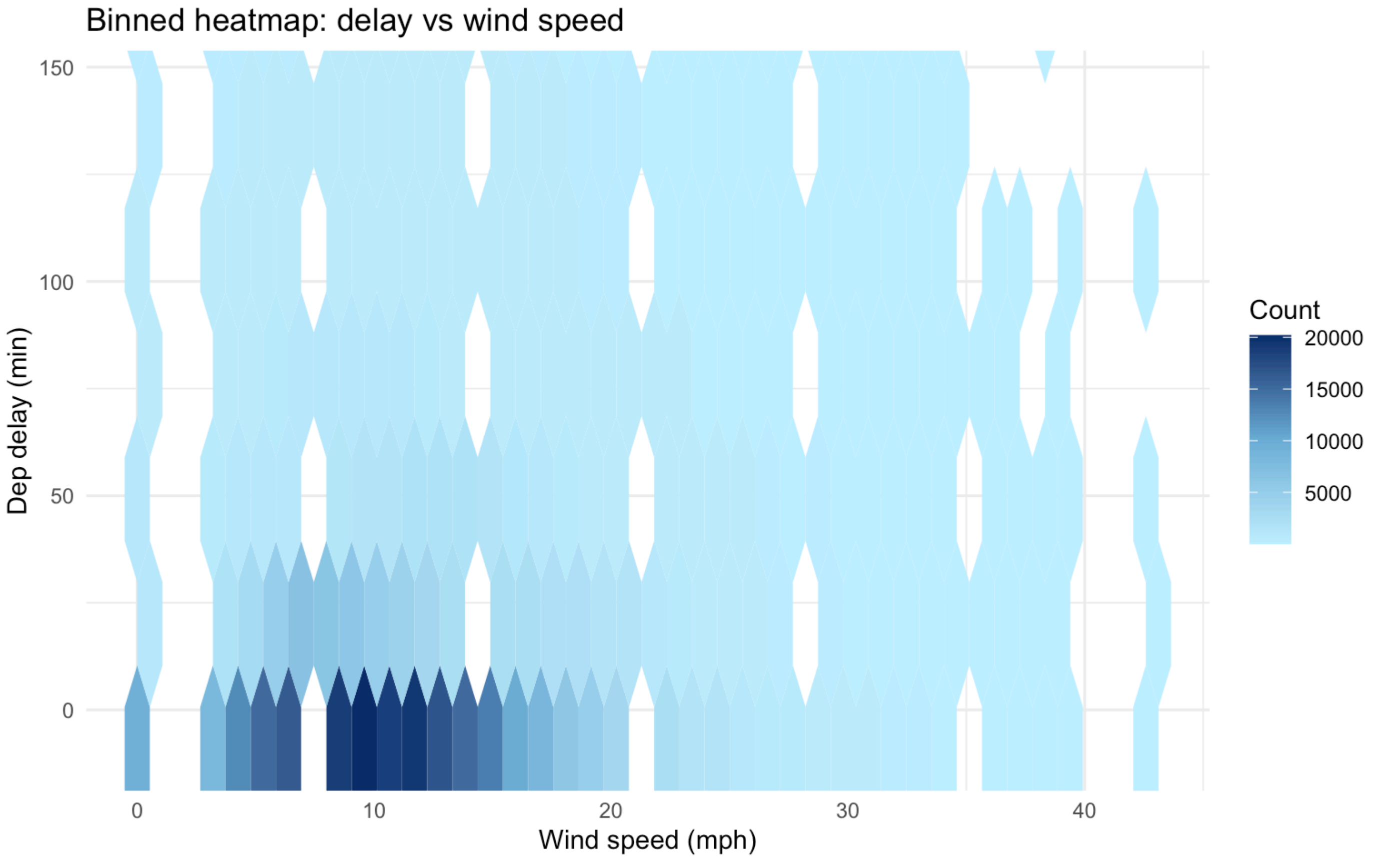
```
ggplot(fw_binned, aes(visib_bin, dep_delay)) +
  geom_violin(fill = scales::alpha(col_visib, .6), color = NA, na.rm = TRUE) +
  geom_boxplot(width = .15, outlier.size = .5, fill = "purple4", na.rm = TRUE) +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Violin + box: delay by visibility", x = "Visibility bin", y = "Dep delay
  theme_minimal()
```



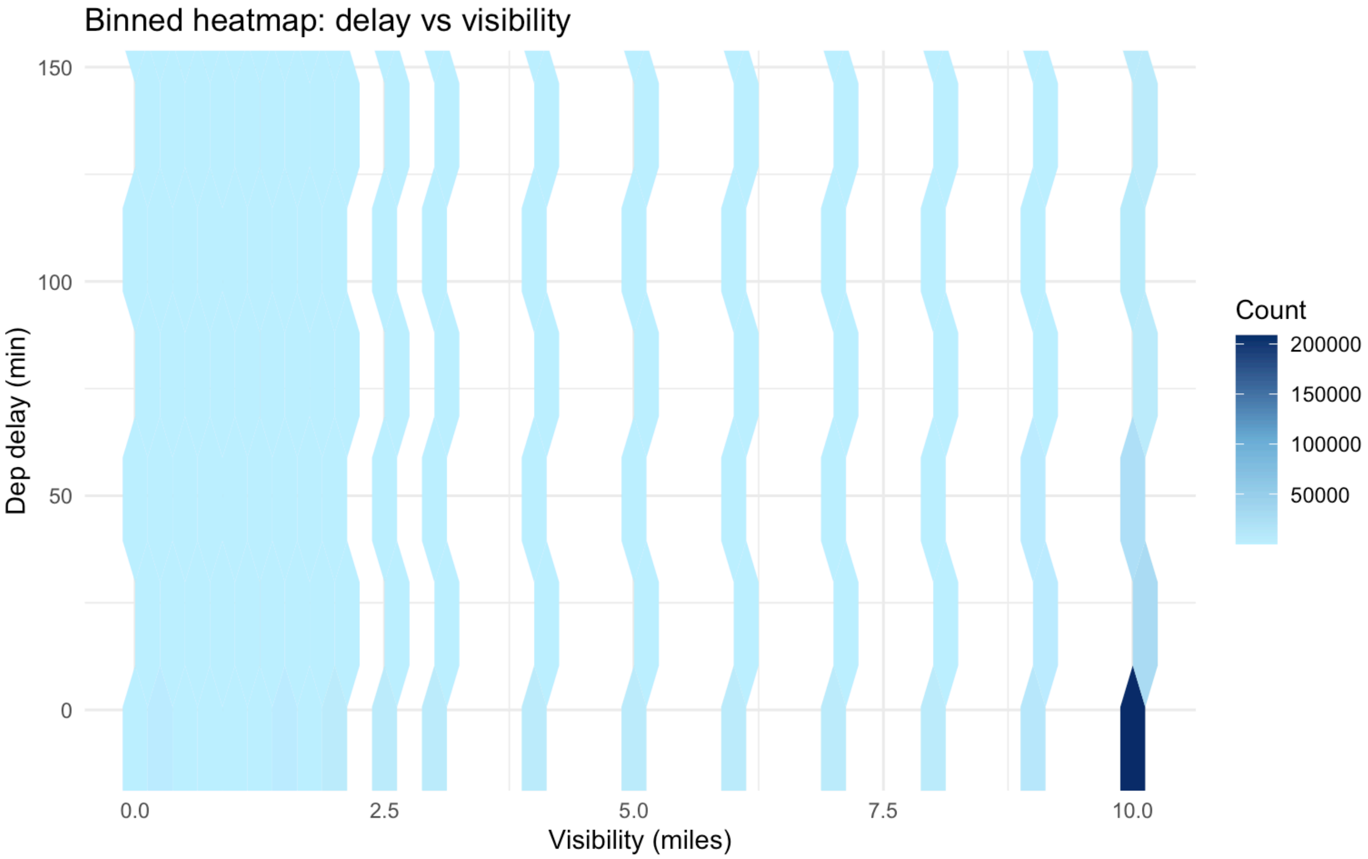
```
# Hexbin
ggplot(flights_weather, aes(precip, dep_delay)) +
  geom_hex(bins = 40, na.rm = TRUE) +
  scale_fill_gradientn(colors = col_bins, name = "Count") +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Binned heatmap: delay vs precipitation", x = "Precip (in)", y = "Dep dela
  theme_minimal()
```



```r
ggplot(flights_weather, aes(wind_speed, dep_delay)) +
  geom_hex(bins = 40, na.rm = TRUE) +
  scale_fill_gradientn(colors = col_bins, name = "Count") +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Binned heatmap: delay vs wind speed", x = "Wind speed (mph)", y = "Dep de
  theme_minimal()
```

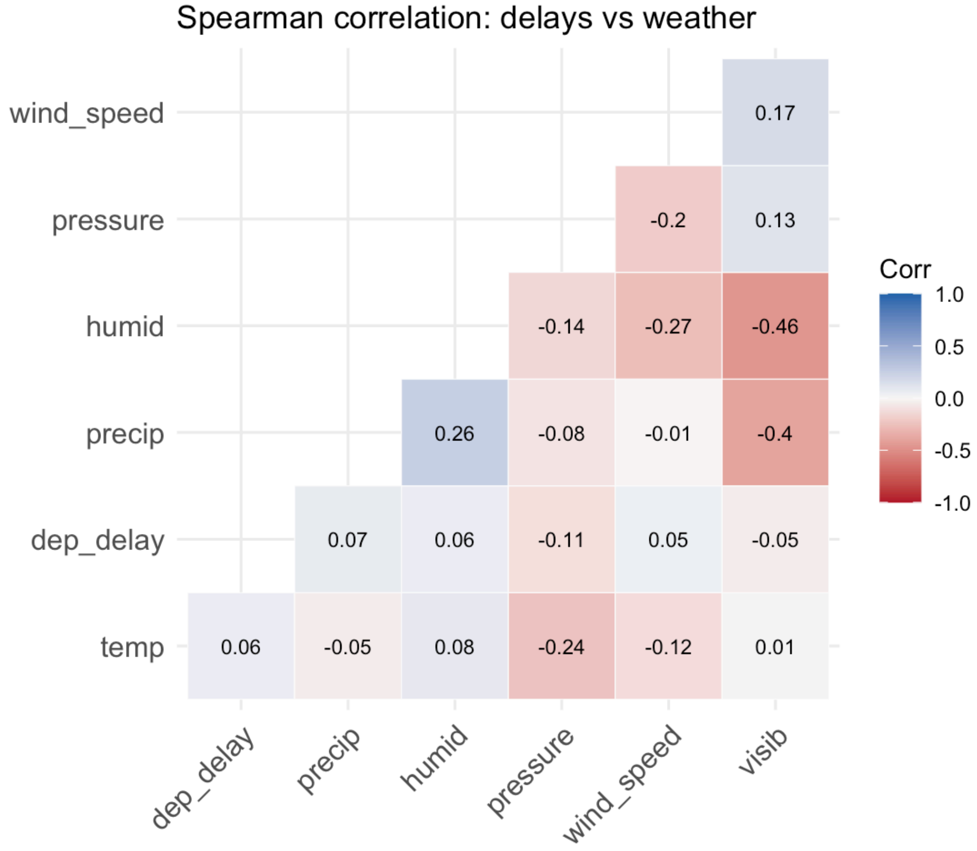


```r
ggplot(flights_weather, aes(visib, dep_delay)) +
  geom_hex(bins = 40, na.rm = TRUE) +
  scale_fill_gradientn(colors = col_bins, name = "Count") +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Binned heatmap: delay vs visibility", x = "Visibility (miles)", y = "Dep
  theme_minimal()
```



```r
# select varaibles
vars <- c("dep_delay", "precip", "wind_speed", "visib", "temp", "humid", "pressure")
mat  <- flights_weather %>%
  select(all_of(vars)) %>%
  mutate(across(everything(), as.numeric)) %>%
  cor(use = "complete.obs", method = "spearman")  # Spearman more robust

# using ggcorrplot
ggcorrplot::ggcorrplot(
  mat, hc.order = TRUE, type = "lower", lab = TRUE, outline.col = "white",
  colors = col_corr, lab_size = 3
) +
  ggtitle("Spearman correlation: delays vs weather")
```

```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
ℹ Please use tidy evaluation idioms with `aes()`.
ℹ See also `vignette("ggplot2-in-packages")` for more information.
ℹ The deprecated feature was likely used in the ggcorrplot package.
  Please report the issue at <https://github.com/kassambara/ggcorrplot/issues>.
```

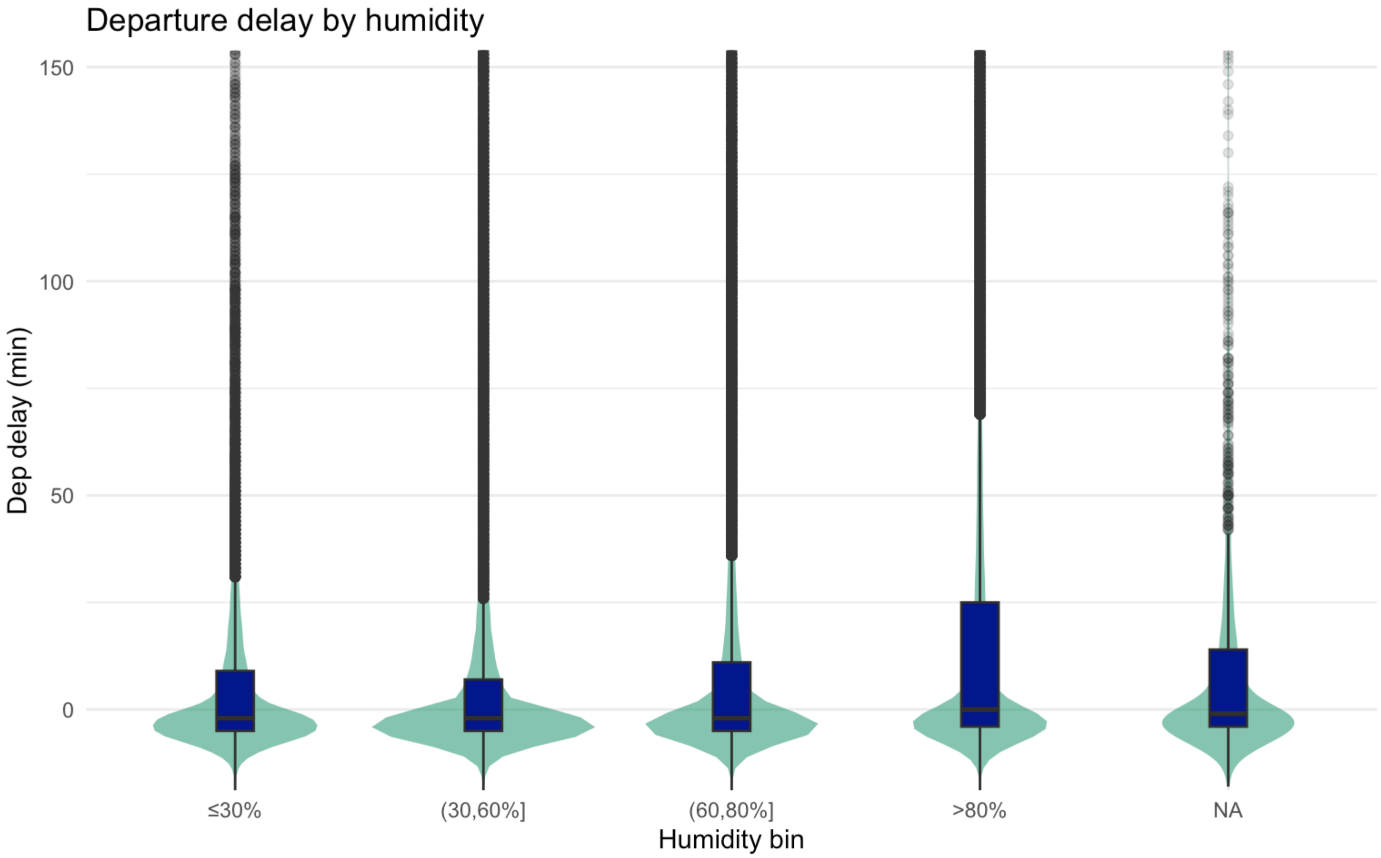


```r
# Colors for selection
col_humid   <- "#1b9e77"   # humidity color
col_press   <- "#d95f02"   # pressure color
col_points  <- "grey35"    # scatter points
col_smooth  <- "#7570b3"   # smooth line

# Trim extreme delays just for plotting readability
q_lim <- quantile(flights_weather$dep_delay, c(.02, .98), na.rm = TRUE)
```

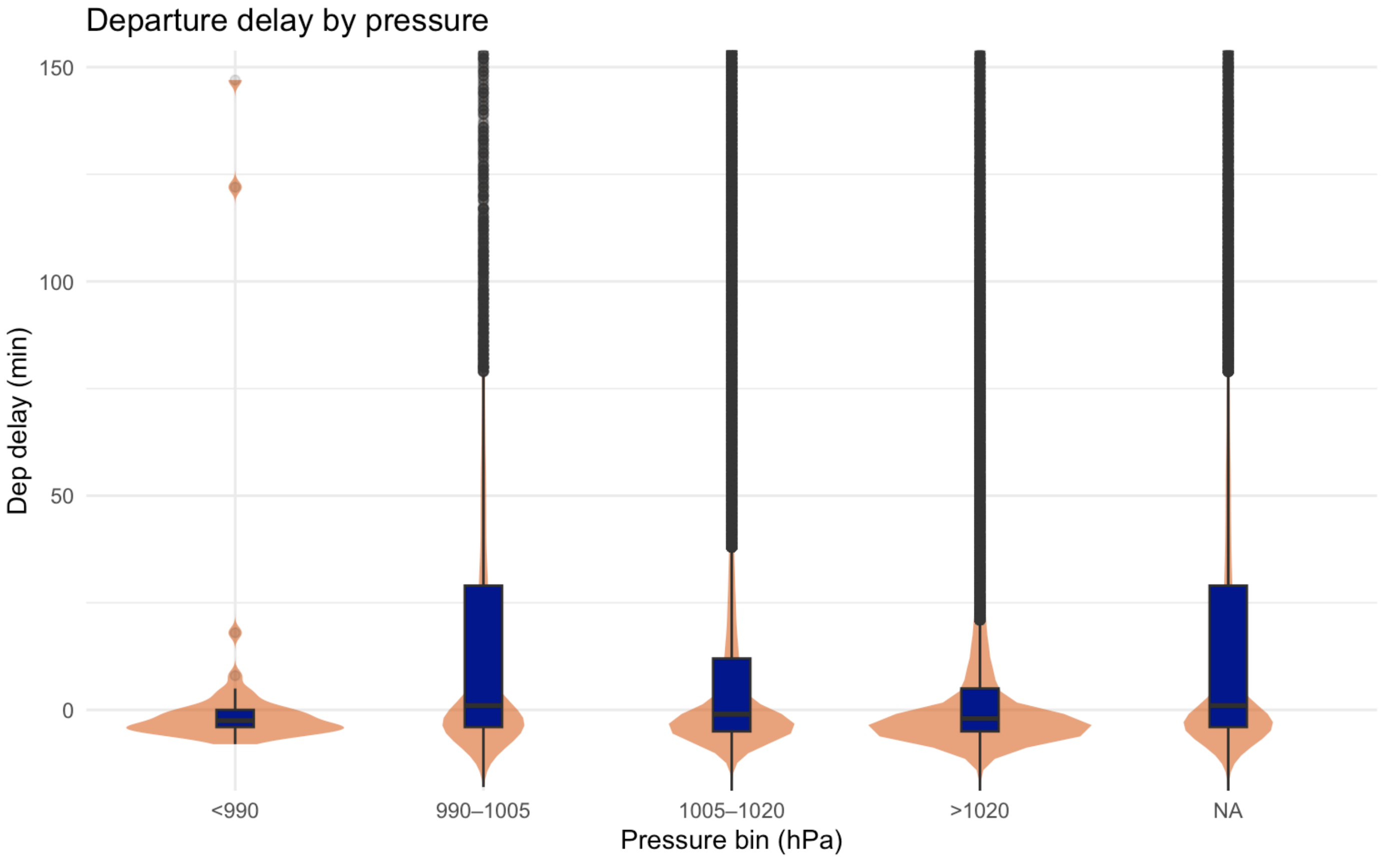
```r
fw_hp <- flights_weather %>%
  mutate(
    # Humidity ranges 0-100 (%). Adjust bins if you prefer.
    humid_bin = cut(humid,
                    breaks = c(-Inf, 30, 60, 80, Inf),
                    labels = c("≤30%", "(30,60%]", "(60,80%]", ">80%")),
    # Pressure is in millibars (hPa) in nycflights13; typical ~980-1040.
    pressure_bin = cut(pressure,
                       breaks = c(-Inf, 990, 1005, 1020, Inf),
                       labels = c("<990", "990-1005", "1005-1020", ">1020"))
  )

sum_humid <- fw_hp %>%
  group_by(humid_bin) %>%
  summarise(mean_dep_delay = mean(dep_delay, na.rm = TRUE),
            n = n(), .groups = "drop") %>%
  arrange(desc(mean_dep_delay))

sum_press <- fw_hp %>%
  group_by(pressure_bin) %>%
  summarise(mean_dep_delay = mean(dep_delay, na.rm = TRUE),
            n = n(), .groups = "drop") %>%
  arrange(desc(mean_dep_delay))

sum_humid; sum_press  # look at which bins have larger delays
```

```
# A tibble: 5 × 3
  humid_bin mean_dep_delay      n
  <fct>              <dbl>  <int>
1 >80%               22.4   65559
2 <NA>               13.5    1573
3 (60,80%]           13.2   89383
4 ≤30%                9.30  16859
5 (30,60%]            8.95 163402

# A tibble: 5 × 3
  pressure_bin mean_dep_delay      n
  <fct>                 <dbl>  <int>
1 <NA>                  24.7   38788
2 990–1005              23.6   11667
3 1005–1020             13.0  174793
4 >1020                  7.01 111482
5 <990                   4.71     46
```

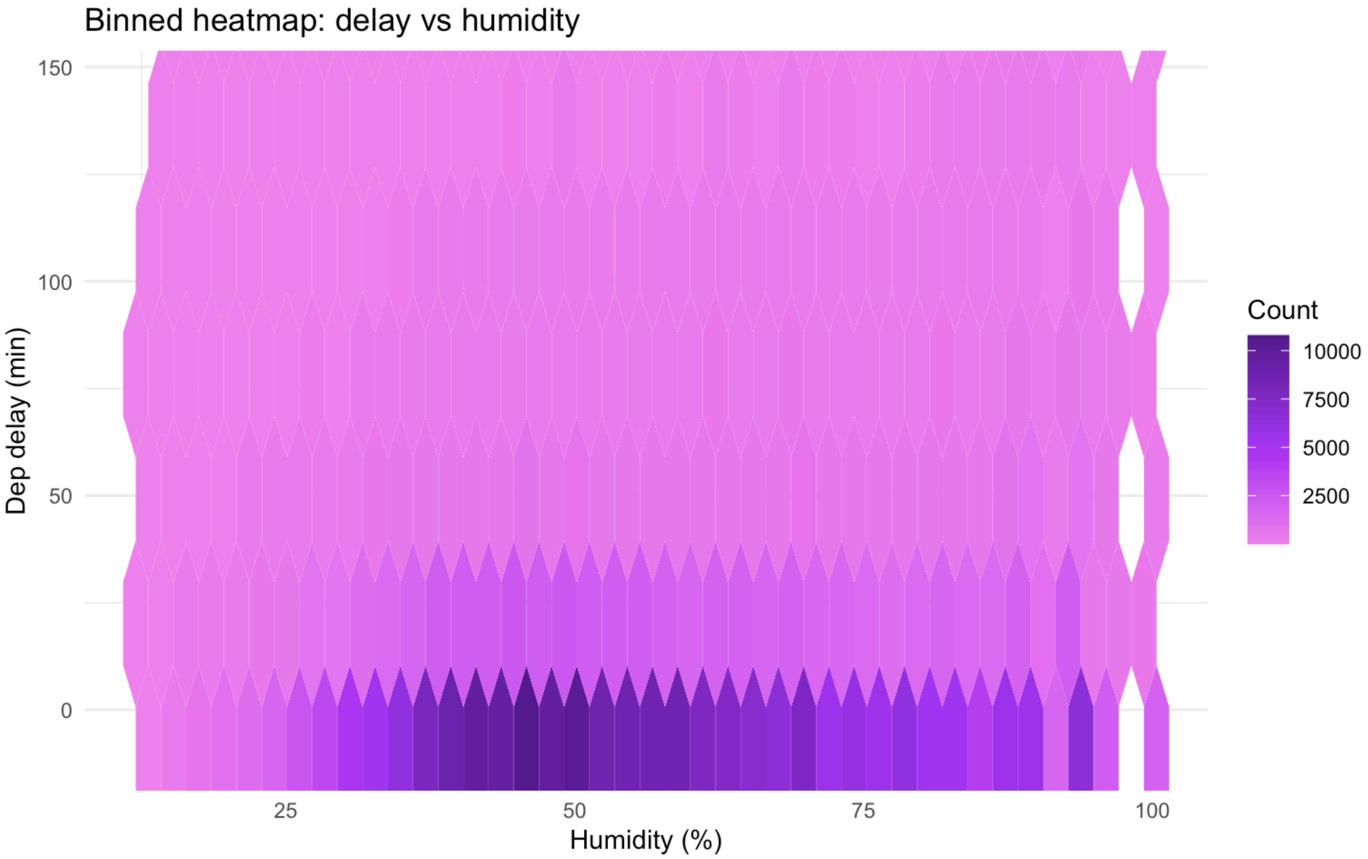
```
# Humidity
ggplot(fw_hp, aes(humid_bin, dep_delay)) +
  geom_violin(fill = scales::alpha(col_humid, .6), color = NA, na.rm = TRUE) +
  geom_boxplot(width = .15, fill = "darkblue", outlier.alpha = .15, na.rm = TRUE) +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Departure delay by humidity", x = "Humidity bin", y = "Dep delay (min)")
  theme_minimal()
```



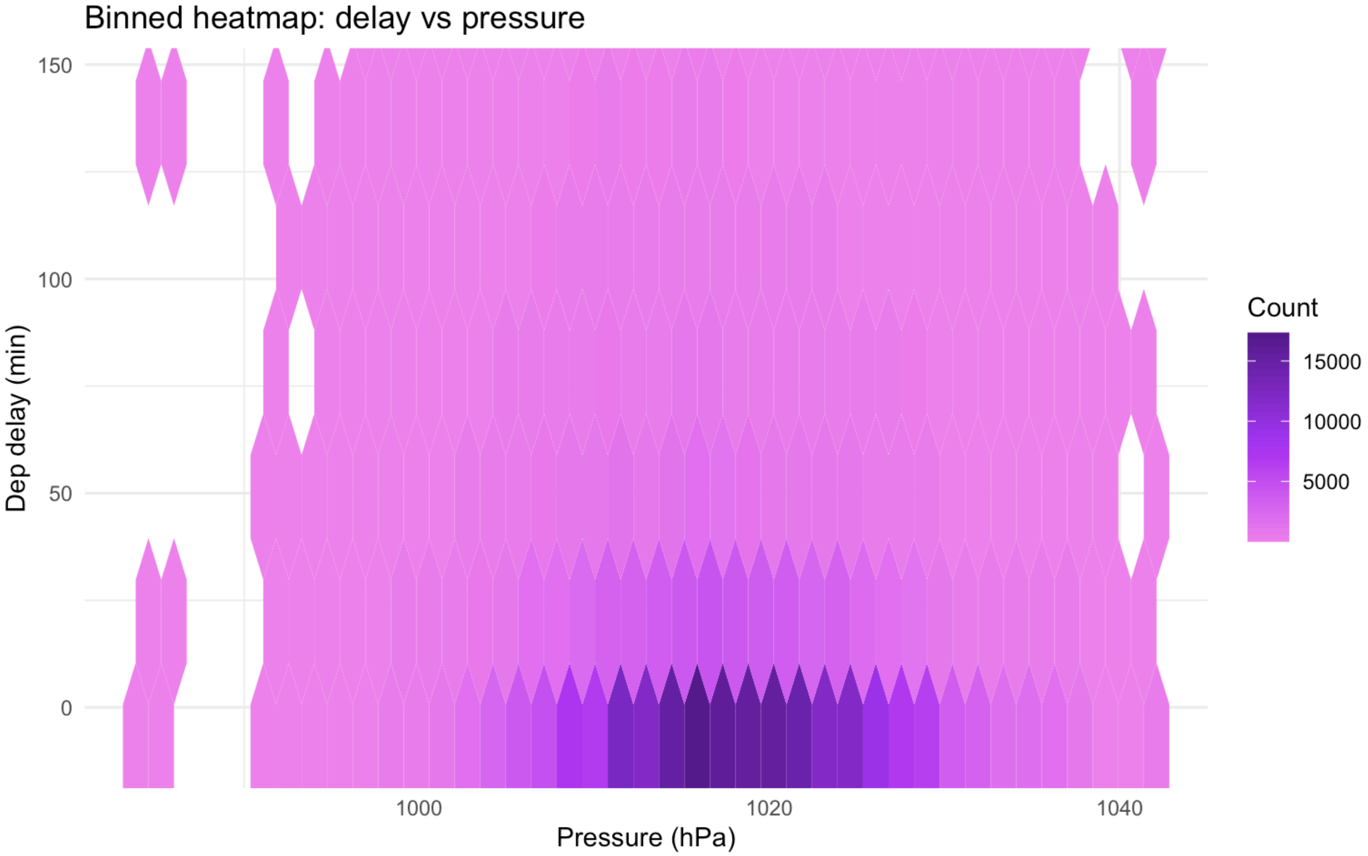
```
# Pressure
ggplot(fw_hp, aes(pressure_bin, dep_delay)) +
  geom_violin(fill = scales::alpha(col_press, .6), color = NA, na.rm = TRUE) +
  geom_boxplot(width = .15, fill = "darkblue", outlier.alpha = .15, na.rm = TRUE) +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Departure delay by pressure", x = "Pressure bin (hPa)", y = "Dep delay (m
  theme_minimal()
```



```
# Hexbin heatmaps (stable with huge n)
col_bins2  <- c("violet", "purple", "purple4")          # low→mid→high for heatmaps

ggplot(flights_weather, aes(humid, dep_delay)) +
  geom_hex(bins = 40, na.rm = TRUE) +
  scale_fill_gradientn(colors = col_bins2, name = "Count") +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Binned heatmap: delay vs humidity", x = "Humidity (%)", y = "Dep delay (m
  theme_minimal()
```

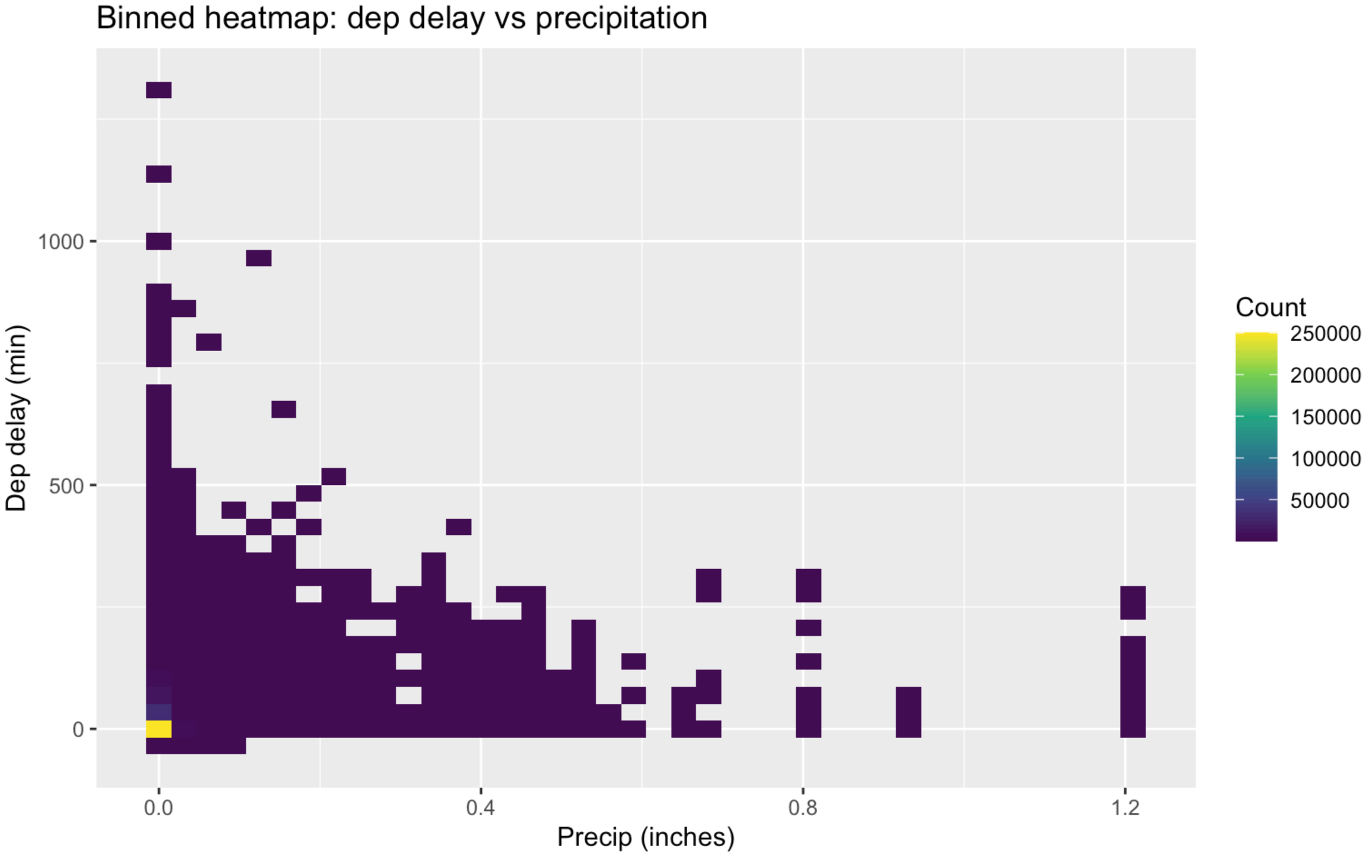


```
ggplot(flights_weather, aes(pressure, dep_delay)) +
  geom_hex(bins = 40, na.rm = TRUE) +
  scale_fill_gradientn(colors = col_bins2, name = "Count") +
  coord_cartesian(ylim = q_lim) +
  labs(title = "Binned heatmap: delay vs pressure", x = "Pressure (hPa)", y = "Dep delay
  theme_minimal()
```



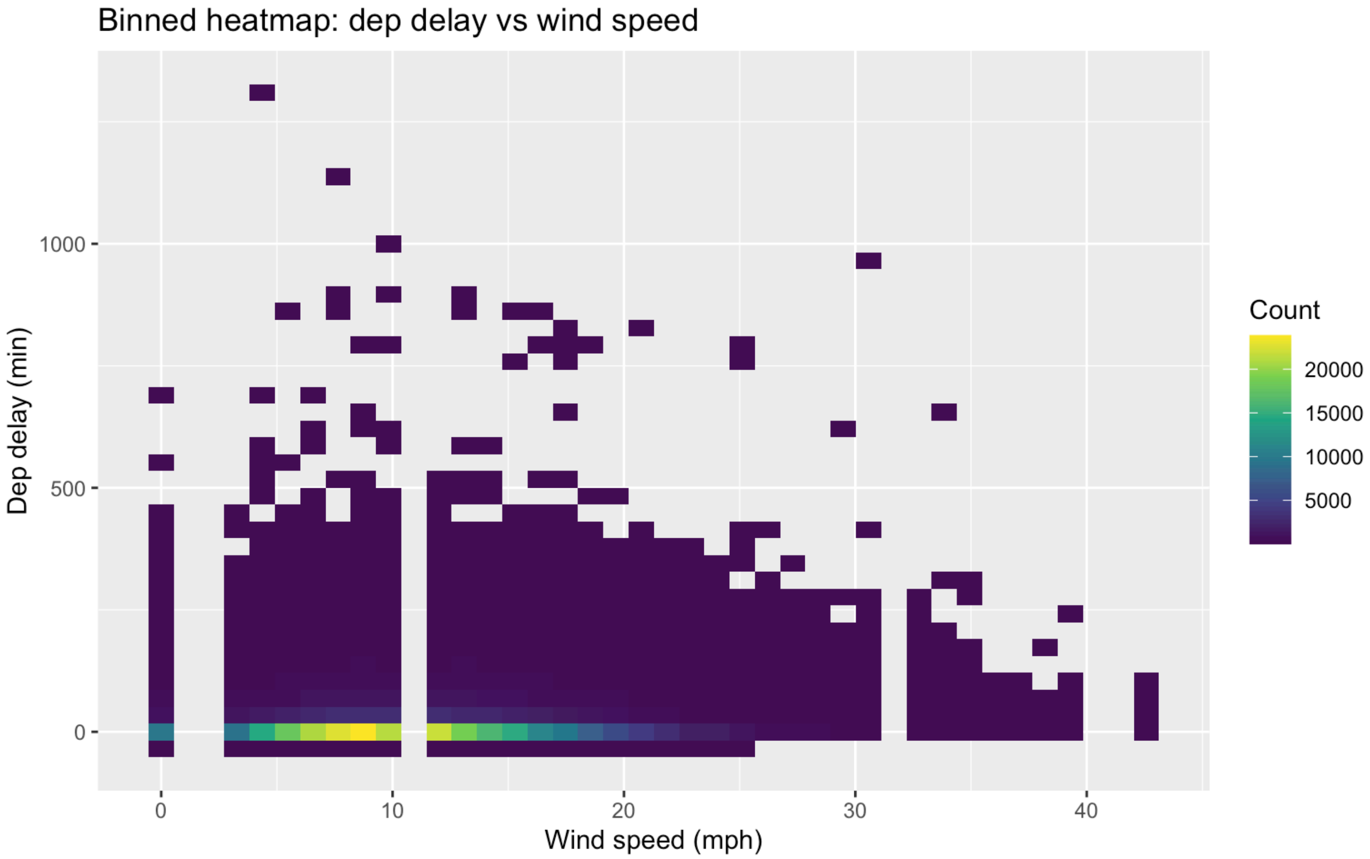
```
# 9d) A quick, robust “top factors” display using partial dependence style smooths
ggplot(flights_weather, aes(precip, dep_delay)) +
  geom_bin2d(bins = 40) +
  scale_fill_viridis_c() +
  labs(title = "Binned heatmap: dep delay vs precipitation",
       x = "Precip (inches)", y = "Dep delay (min)", fill = "Count")
```

```
Warning: Removed 9783 rows containing non-finite outside the scale range
(`stat_bin2d()`).
```



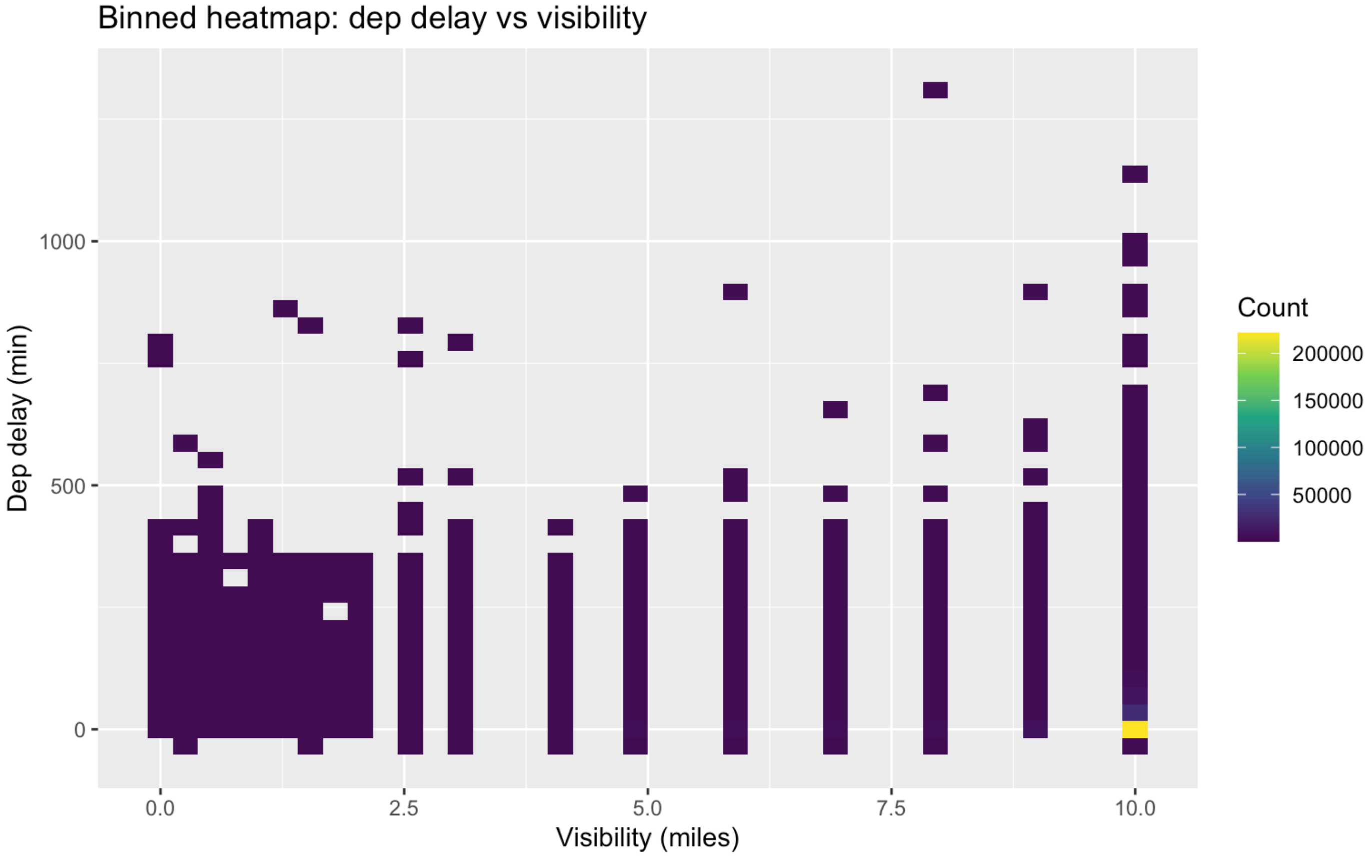
```
ggplot(flights_weather, aes(wind_speed, dep_delay)) +
  geom_bin2d(bins = 40) +
  scale_fill_viridis_c() +
  labs(title = "Binned heatmap: dep delay vs wind speed",
       x = "Wind speed (mph)", y = "Dep delay (min)", fill = "Count")
```

```
Warning: Removed 9861 rows containing non-finite outside the scale range
(`stat_bin2d()`).
```



```r
ggplot(flights_weather, aes(visib, dep_delay)) +
  geom_bin2d(bins = 40) +
  scale_fill_viridis_c() +
  labs(title = "Binned heatmap: dep delay vs visibility",
       x = "Visibility (miles)", y = "Dep delay (min)", fill = "Count")
```

```
Warning: Removed 9783 rows containing non-finite outside the scale range
(`stat_bin2d()`).
```



Reports: In my opinion, I treated “impact” as a shift in the distribution of departure delay, not just referred to single correlation, because delays are heavy-tailed and zero-inflated. Given to such understanding, based on the plots above, precipitation had the strongest impact. The violin-box plot by precipitation bin showed large right-shift and dramatic spread as precipitation increases, with the >1 inch bin having a much higher median and very long upper tail.The mean delay by precipitation bin rises steeply (≈10 min on dry days to ~100+ min when >1 inch). Therefore, precipitation showed a large absolute effect on delay minutes and increases the chance of extreme delays. In summation, heavy precipitation is the dominant driver of longer and more variable departure delays, followed by poor visibility and high winds; humidity, pressure, and temperature show at relatively weak associations.