DataSci 306, Homework 5

Max Han, maxhan

```
knitr::opts_chunk$set(echo = TRUE)
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   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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 errors
```

```
library(lubridate)
library(nycflights13)
```

Reshaping and Joins

In almost all problems it would be easier for you if you reshape your dataframe using pivot_wider and/or pivot_longer and/or use joins

Problem 1 (3 points)

(a) Let us revisit the storms data and let us view the total number of storms that are of category 1 through 5 across each available month as shown in the fig-1.png enclosed with this HW (1 point)

```
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> <fct>           <dbl> <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropical de~       NA    25     1013
## 2 Amy    1975     6    27     6  28.5 -79   tropical de~       NA    25     1013
## 3 Amy    1975     6    27    12  29.5 -79   tropical de~       NA    25     1013
## 4 Amy    1975     6    27    18  30.5 -79   tropical de~       NA    25     1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropical de~       NA    25     1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropical de~       NA    25     1012
## # i 2 more variables: tropicalstorm_force_diameter <int>,
## #   hurricane_force_diameter <int>
```

```
storms |> filter(category %in% c(1:5)) |> group_by(category, month) |> summarise(n = n(), .groups = "dr
```

```
## # A tibble: 5 x 9
##   category   `1`   `6`   `7`   `8`   `9`  `10`  `11`  `12`
##      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1        1     5    18   148   581  1145   466   152    33
```

```
## 2        2     0     0    35   198   581   150    29     0
## 3        3     0     0    19   113   359    86    16     0
## 4        4     0     0    18   114   309    88    24     0
## 5        5     0     0     1    32    70    13     0     0
```

See fig-1.png that is part of this HW

(b) Iris dataset (2 points) For this problem, we will work on the iris data. To learn more on iris, please read the documentation with `?iris` as always

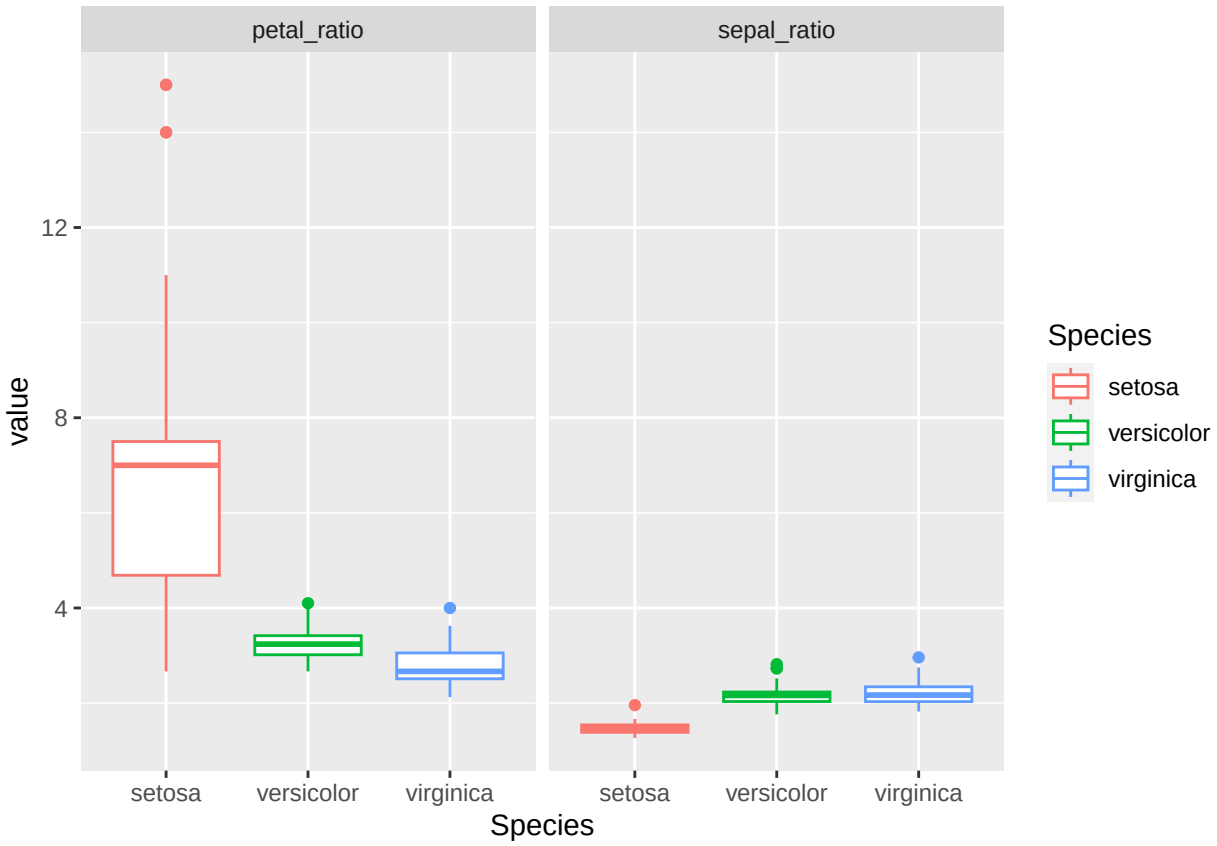
```
iris |> head()
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Add columns to show the ratio of Length/Width for both sepal and petal for the iris dataset such that the sepal_ratio shows the ratio of Sepal.Length/Sepal.Width and the petal_ratio shows the ratio of Petal.Length/Petal.Width. Then reshape the dataset to make it longer with only three columns; Species, ratio and value; where ratio column contains both petal_ratio and sepal_ratio and then reproduce the following plot. (2 points)

See ratio.png that is part of this HW

```
iris |> mutate(sepal_ratio = Sepal.Length / Sepal.Width, petal_ratio = Petal.Length / Petal.Width) |>
  pivot_longer(cols = c(sepal_ratio, petal_ratio), names_to = "ratio", values_to = "value") |>
  select(Species, ratio, value) |> ggplot(aes(y = value, x = Species, color = Species)) +
  geom_boxplot() + facet_wrap(~ratio)
```



Problem 2 (4 points)

(a) Age of aircraft vs Cancellations (2 points)

We will look into the `flights` dataset again for this problem

Is there a relation between age of the aircraft and proportion of cancellations? Answer this question using a suitable chart

Hint: This data was collected in 2013. Derive the age of the aircraft using the year of manufacture of the aircraft and then find the proportion of flights cancelled.

```
flights |> print()
```

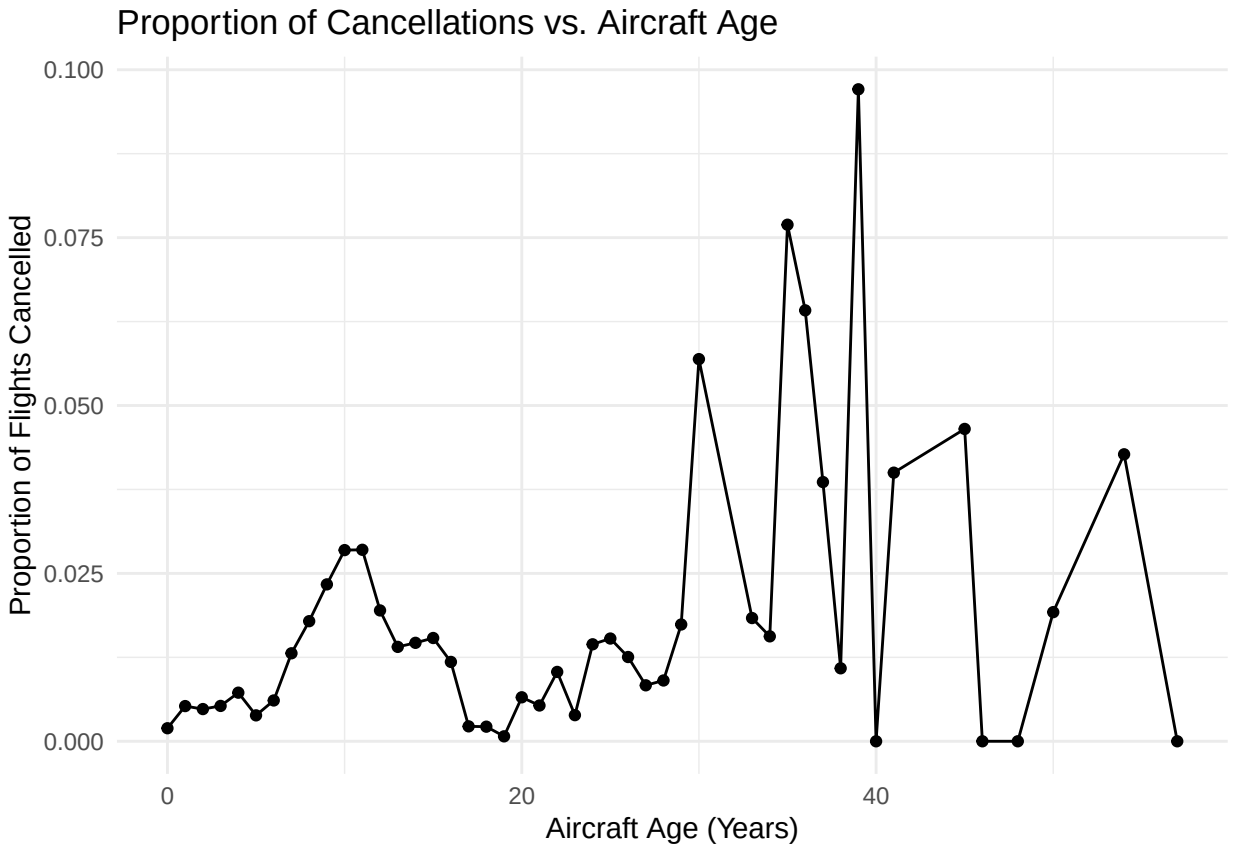
```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # i 336,766 more rows
## # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
```

```
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
flights_age <- flights |>
  left_join(planes, by = "tailnum") |>
  filter(!is.na(year.y)) |>
  mutate(age = 2013 - as.numeric(year.y), .before = year.x)

flights_cancellation <- flights_age %>%
  group_by(age) %>%
  summarise(total_flights = n(),
            cancellations = sum(is.na(dep_time)),
            prop_cancelled = cancellations / total_flights)

ggplot(flights_cancellation, aes(x = age, y = prop_cancelled)) +
  geom_point() +
  geom_line() +
  labs(title = "Proportion of Cancellations vs. Aircraft Age",
       x = "Aircraft Age (Years)",
       y = "Proportion of Flights Cancelled") +
  theme_minimal()
```




```
flights_cancellation
```

```
## # A tibble: 46 x 4
##      age total_flights cancellations prop_cancelled
##    <dbl>         <int>         <int>          <dbl>
##  1     0          4630             9        0.00194
```

```
##  2     1          7252           38        0.00524
##  3     2          6046           29        0.00480
##  4     3          3797           20        0.00527
##  5     4          6632           48        0.00724
##  6     5         17878           69        0.00386
##  7     6         15300           93        0.00608
##  8     7         13203          173        0.0131
##  9     8         14369          257        0.0179
## 10     9         15706          367        0.0234
## # i 36 more rows
```

The graphs shows no pattern. Thus, we can conclude that there is no relationship between plane age and the proportion of cancellation.

(b) Dep Delay due to weather? (2 points)

Filter all the flights that have a departure delay of more than 60 minutes, and then find if weather parameters are related to delays. The weather parameters to consider are humid, wind_speed, wind_gust, precip, pressure, visib. Using a facet_grid, and a suitable geom, slice and dice the dataset based on the three origins and the given weather factors and explain which of these weather parameters influence delays if any.

Hint: You may want to join along with reshape to get the end result.

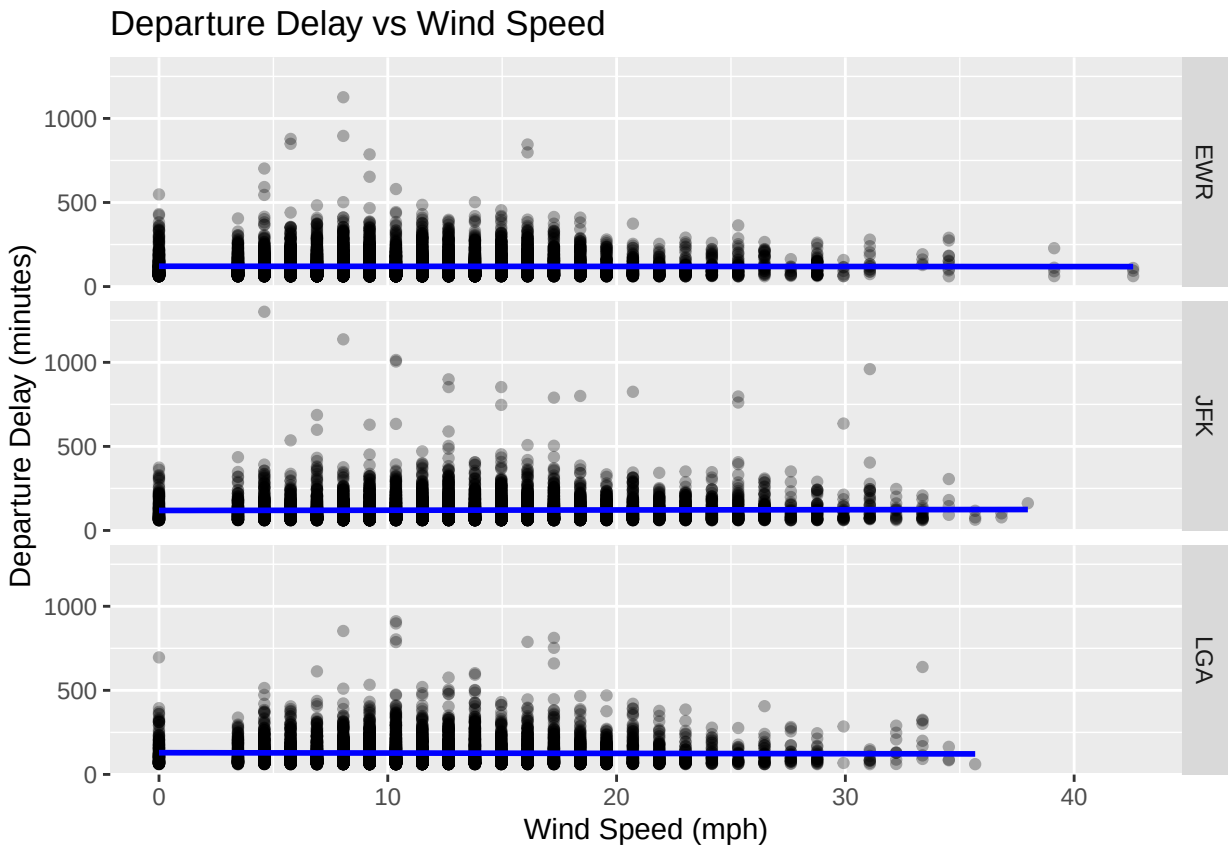
```
delayed_flights_weather <- flights |> filter(dep_delay > 60) |> left_join(weather, by = c("origin", "yea

ggplot(delayed_flights_weather, aes(x = wind_speed, y = dep_delay)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "blue") +
  facet_grid(origin ~ .) +
  labs(title = "Departure Delay vs Wind Speed",
       x = "Wind Speed (mph)",
       y = "Departure Delay (minutes)")
```

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

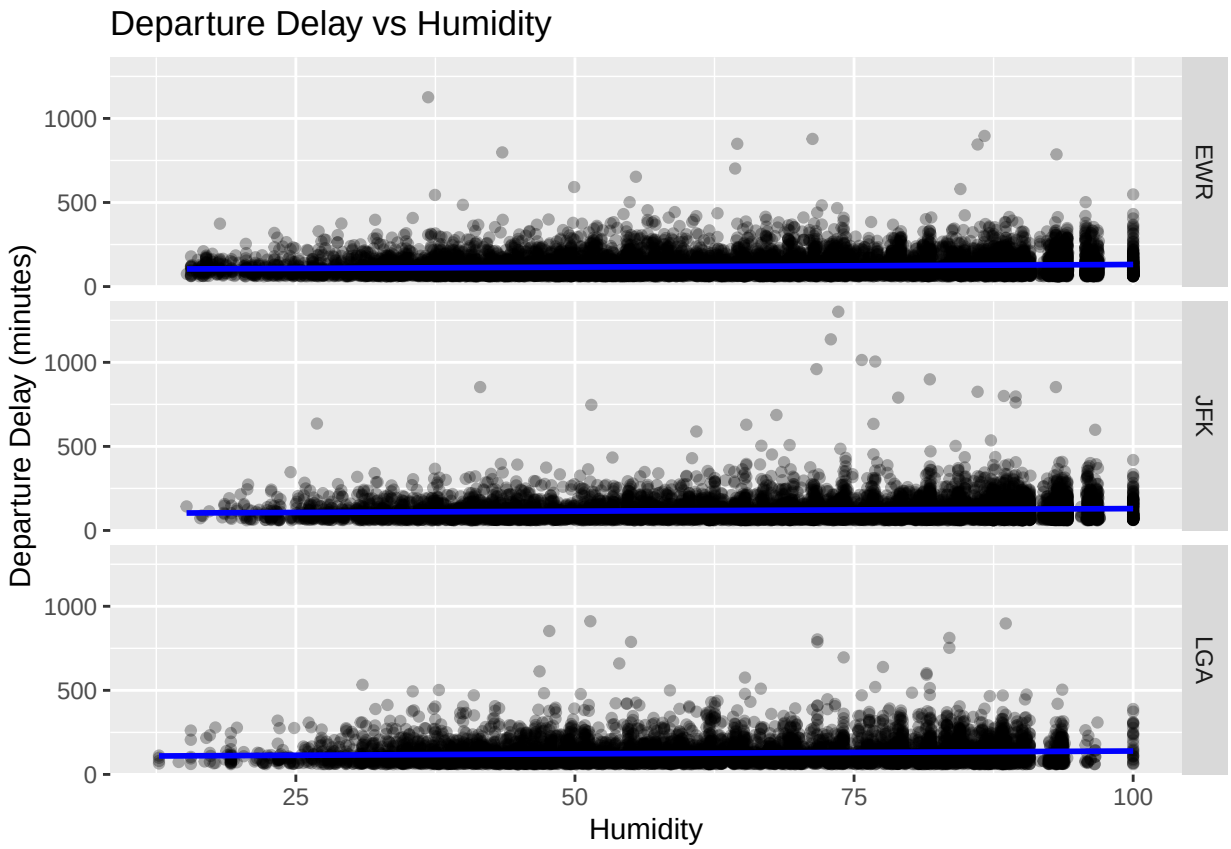
```
## Warning: Removed 134 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 134 rows containing missing values (`geom_point()`).
```




```
ggplot(delayed_flights_weather, aes(x = humid, y = dep_delay)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "blue") +
  facet_grid(origin ~ .) +
  labs(title = "Departure Delay vs Humidity",
       x = "Humidity",
       y = "Departure Delay (minutes)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 132 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 132 rows containing missing values (`geom_point()`).
```

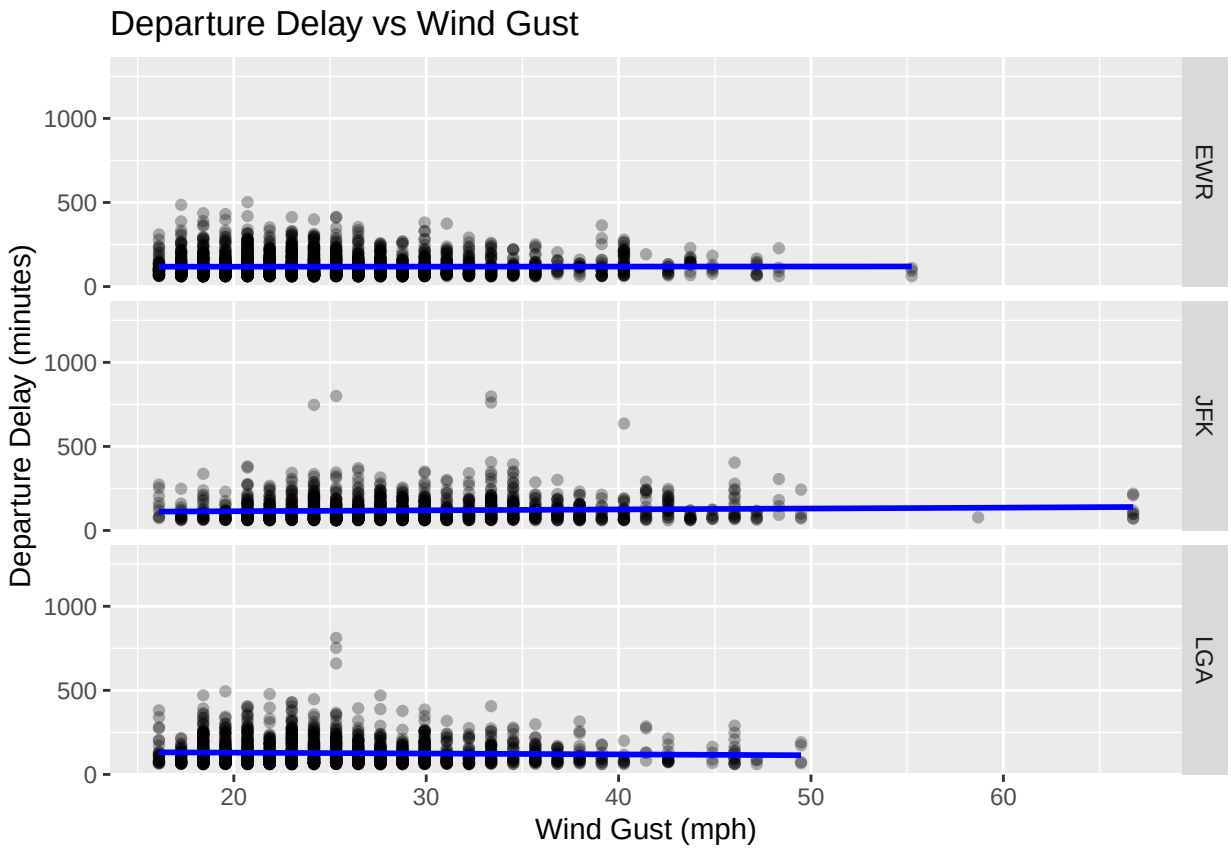



```
ggplot(delayed_flights_weather, aes(x = wind_gust, y = dep_delay)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "blue") +
  facet_grid(origin ~ .) +
  labs(title = "Departure Delay vs Wind Gust",
       x = "Wind Gust (mph)",
       y = "Departure Delay (minutes)")
```

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

```
## Warning: Removed 19965 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 19965 rows containing missing values (`geom_point()`).
```

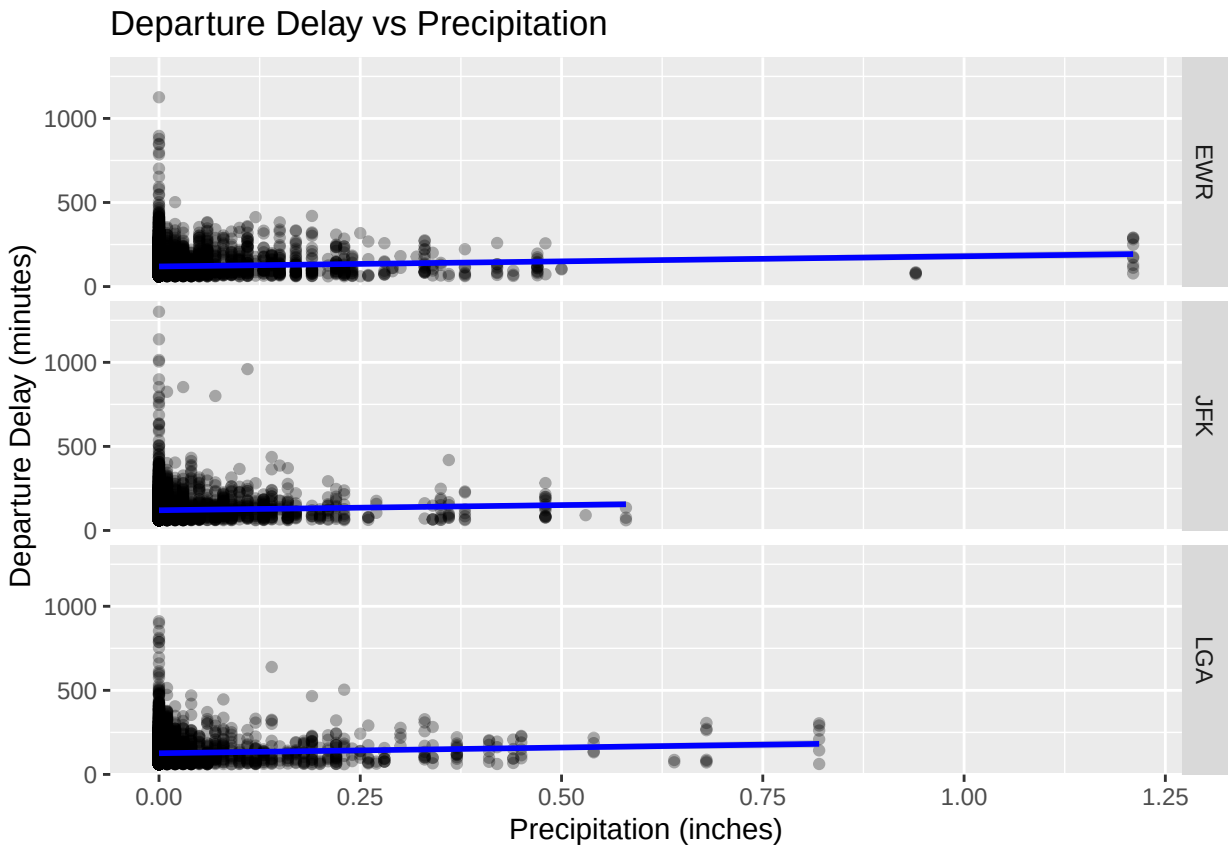



```
ggplot(delayed_flights_weather, aes(x = precip, y = dep_delay)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "blue") +
  facet_grid(origin ~ .) +
  labs(title = "Departure Delay vs Precipitation",
       x = "Precipitation (inches)",
       y = "Departure Delay (minutes)")
```

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

```
## Warning: Removed 131 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 131 rows containing missing values (`geom_point()`).
```

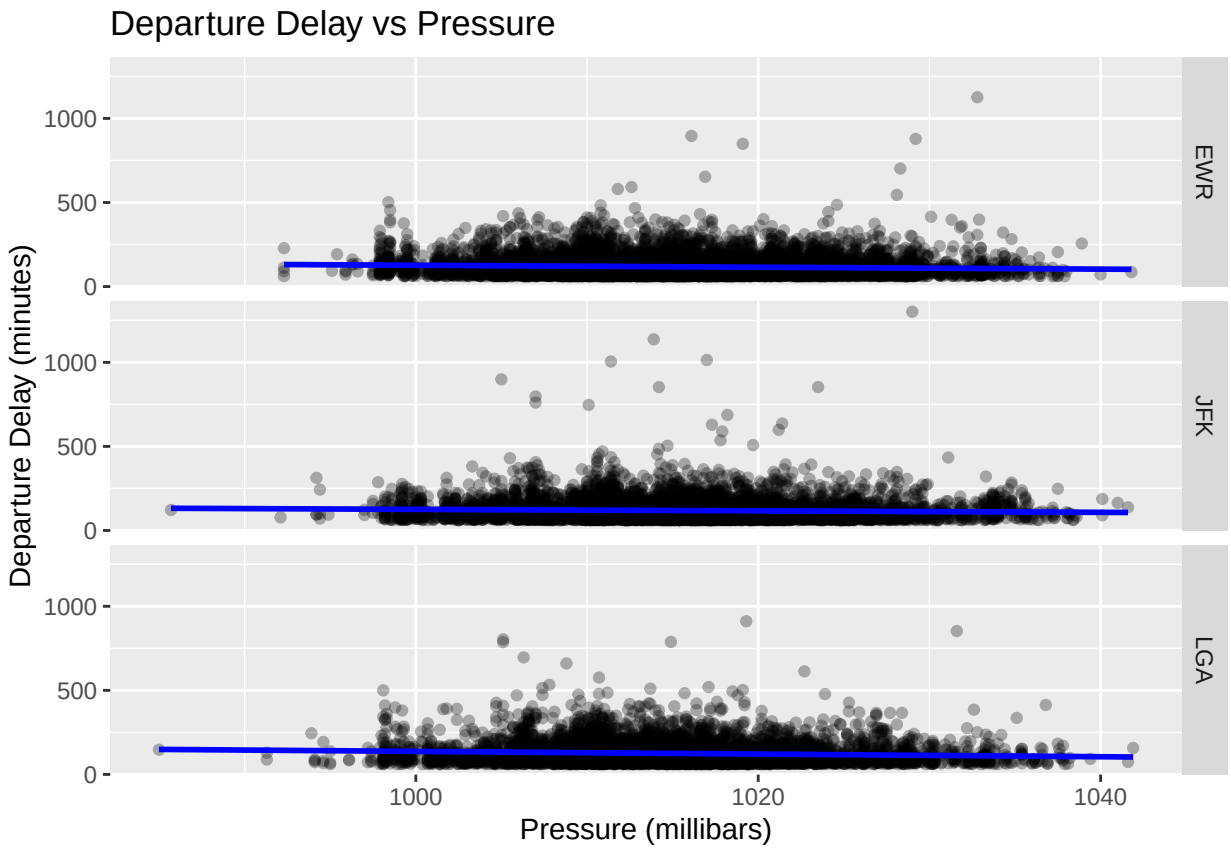



```
ggplot(delayed_flights_weather, aes(x = pressure, y = dep_delay)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "blue") +
  facet_grid(origin ~ .) +
  labs(title = "Departure Delay vs Pressure",
       x = "Pressure (millibars)",
       y = "Departure Delay (minutes)")
```

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

```
## Warning: Removed 5489 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 5489 rows containing missing values (`geom_point()`).
```

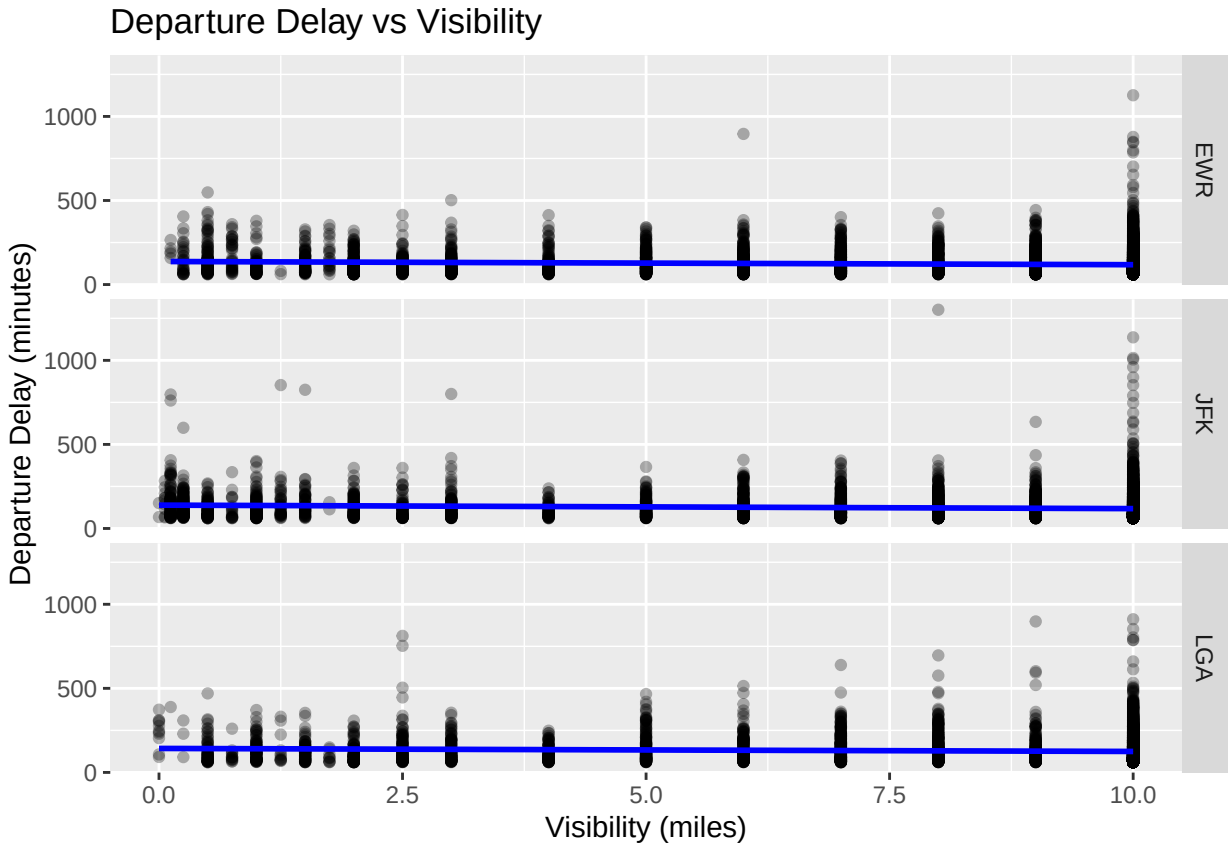



```
ggplot(delayed_flights_weather, aes(x = visib, y = dep_delay)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "blue") +
  facet_grid(origin ~ .) +
  labs(title = "Departure Delay vs Visibility",
       x = "Visibility (miles)",
       y = "Departure Delay (minutes)")
```

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

```
## Warning: Removed 131 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 131 rows containing missing values (`geom_point()`).
```



All the plots shows flat line for linear relationship, which means that slopes are 0. Thus, we can conclude weather doesn't affect delay time.

Problem 3 (3 points)

In this problem, we will explore the `relig_income` dataset. As always learn more on this dataset by running `?relig_income`

```
relig_income |> head()
```

```
## # A tibble: 6 x 11
##   religion  `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k` `$50-75k` `$75-100k`
##   <chr>       <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>       <dbl>
## 1 Agnostic       27        34        60        81        76       137         122
## 2 Atheist        12        27        37        52        35        70          73
## 3 Buddhist       27        21        30        34        33        58          62
## 4 Catholic      418       617       732       670       638      1116         949
## 5 Don't kn~      15        14        15        11        10        35          21
## 6 Evangeli~     575       869      1064       982       881      1486         949
## # i 3 more variables: `$100-150k` <dbl>, `>150k` <dbl>,
## #   `Don't know/refused` <dbl>
```

When you look at the dataset, you will notice that this data is not tidy. First make the data tidy by creating new columns called 'income_range' and 'count' to rearrange the existing column names.

Once that step is done, filter out the category 'Don't know/refused' category that is present in both religion and income_range and then get all the religions that have a total count across all income groups greater or equal to 200 and show the distribution of these religions across the various income groups using a suitable plot. Show a distribution chart that makes it simple to compare religious distribution across income levels.

```r
joined <- relig_income |> pivot_longer(-religion, names_to = "income_range", values_to = "count")
joined
```

```
## # A tibble: 180 x 3
##    religion income_range       count
##    <chr>    <chr>              <dbl>
##  1 Agnostic <$10k                 27
##  2 Agnostic $10-20k               34
##  3 Agnostic $20-30k               60
##  4 Agnostic $30-40k               81
##  5 Agnostic $40-50k               76
##  6 Agnostic $50-75k              137
##  7 Agnostic $75-100k             122
##  8 Agnostic $100-150k            109
##  9 Agnostic >150k                 84
## 10 Agnostic Don't know/refused    96
## # i 170 more rows
```

```r
cleaned <- joined |> filter(income_range != "Don't know/refused" & count != "Don't know/refused")
cleaned
```

```
## # A tibble: 162 x 3
##    religion income_range count
##    <chr>    <chr>        <dbl>
##  1 Agnostic <$10k           27
##  2 Agnostic $10-20k         34
##  3 Agnostic $20-30k         60
##  4 Agnostic $30-40k         81
##  5 Agnostic $40-50k         76
##  6 Agnostic $50-75k        137
##  7 Agnostic $75-100k       122
##  8 Agnostic $100-150k      109
##  9 Agnostic >150k           84
## 10 Atheist  <$10k           12
## # i 152 more rows
```

```r
over_200_count_religion <- cleaned |> group_by(religion) |> summarise(count_over_200 = sum(count, na.rm
over_200_count_religion
```

```
## # A tibble: 13 x 1
##    religion
##    <chr>
##  1 Agnostic
##  2 Atheist
##  3 Buddhist
##  4 Catholic
##  5 Evangelical Prot
##  6 Hindu
##  7 Historically Black Prot
##  8 Jewish
##  9 Mainline Prot
## 10 Mormon
## 11 Orthodox
## 12 Other Faiths
## 13 Unaffiliated
```

```
filtered_income <- cleaned |> inner_join(over_200_count_religion, by = "religion")
filtered_income
```

```
## # A tibble: 117 x 3
##    religion income_range count
##    <chr>    <chr>        <dbl>
##  1 Agnostic <$10k           27
##  2 Agnostic $10-20k         34
##  3 Agnostic $20-30k         60
##  4 Agnostic $30-40k         81
##  5 Agnostic $40-50k         76
##  6 Agnostic $50-75k        137
##  7 Agnostic $75-100k       122
##  8 Agnostic $100-150k      109
##  9 Agnostic >150k           84
## 10 Atheist  <$10k           12
## # i 107 more rows
```

```
filtered_income |> ggplot(aes(x = income_range, y = count, fill = religion)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of Religions Across Income Levels",
       x = "Income Range",
       y = "Count",
       fill = "Religion")
```