MSc Bioinformatics

Marc Henrion

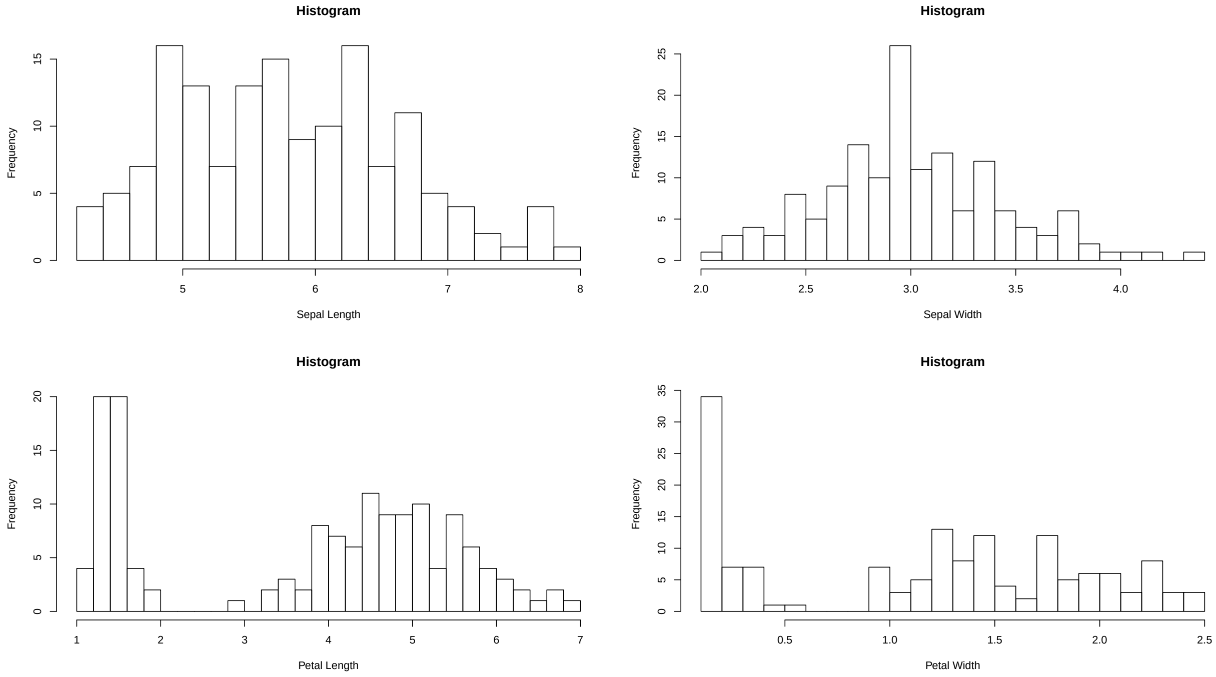
17 January 2020

Session 7 - Practical (Solutions)

1. Using the `iris` dataset (type `?iris` to get more information about this dataset) that comes pre-loaded with R, produce the following figures:
    - Produce histograms for each of `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`. Try using both the graphics (base R) and ggplot2 (tidyverse R) libraries.
    - Produce a bar plot for `Species`.
    - Produce box and whisker plots for each of the 4 continuous variables. Put them all on a single, multi-panel figure.
    - Repeat for just `Sepal.Length` using a violin plot, stratifying by `Species`.
    - Produce a single graph (not multi-panel) that has histograms for `Sepal.Length` for each of the 3 flower species.
    - There are 4 continuous variables. This means there are 6 possible pairs of these. For each such pair, produce a scatter plot of one variable against the other and highlight the different flower species by using a different colour for each species.
    - For one of these 6 scatter plots: estimate the bivariate probability density and add density contour lines to the figure.

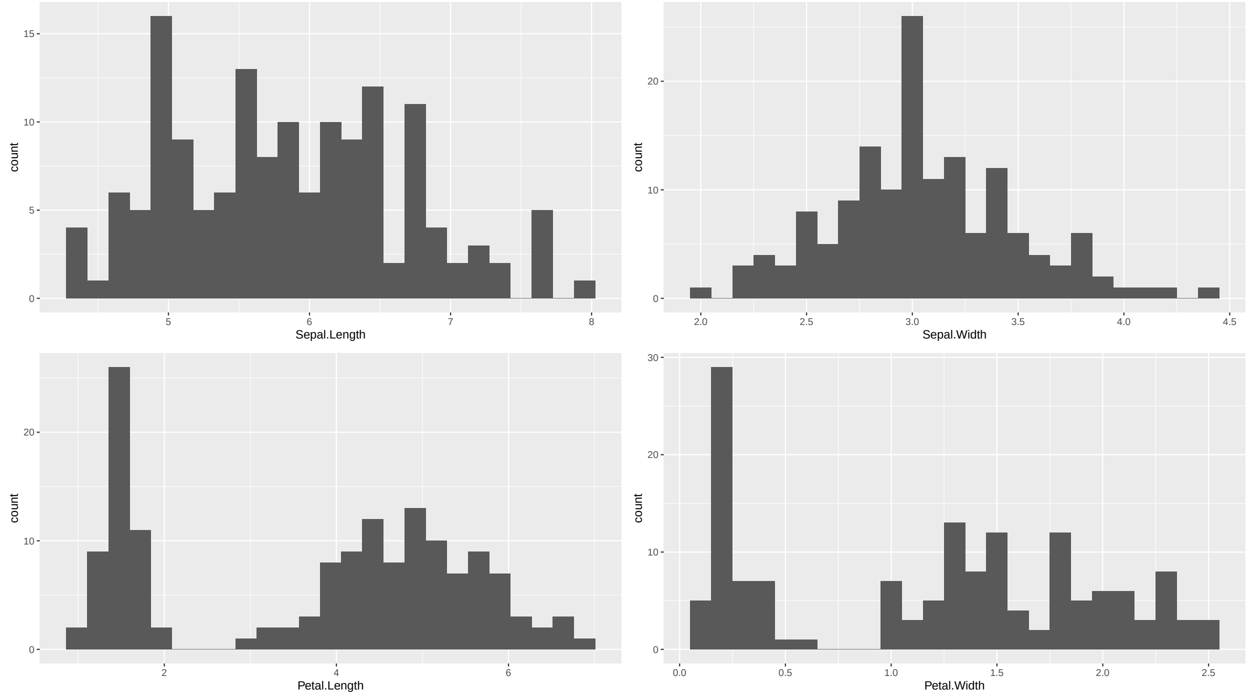
- Histograms using base R:

```r
par(mfrow=c(2,2))
for(i in 1:4){
  hist(iris[,i],
       breaks=25,
       main="Histogram",
       xlab=gsub(pattern="\\.",replacement=" ",colnames(iris)[i]))
}
```



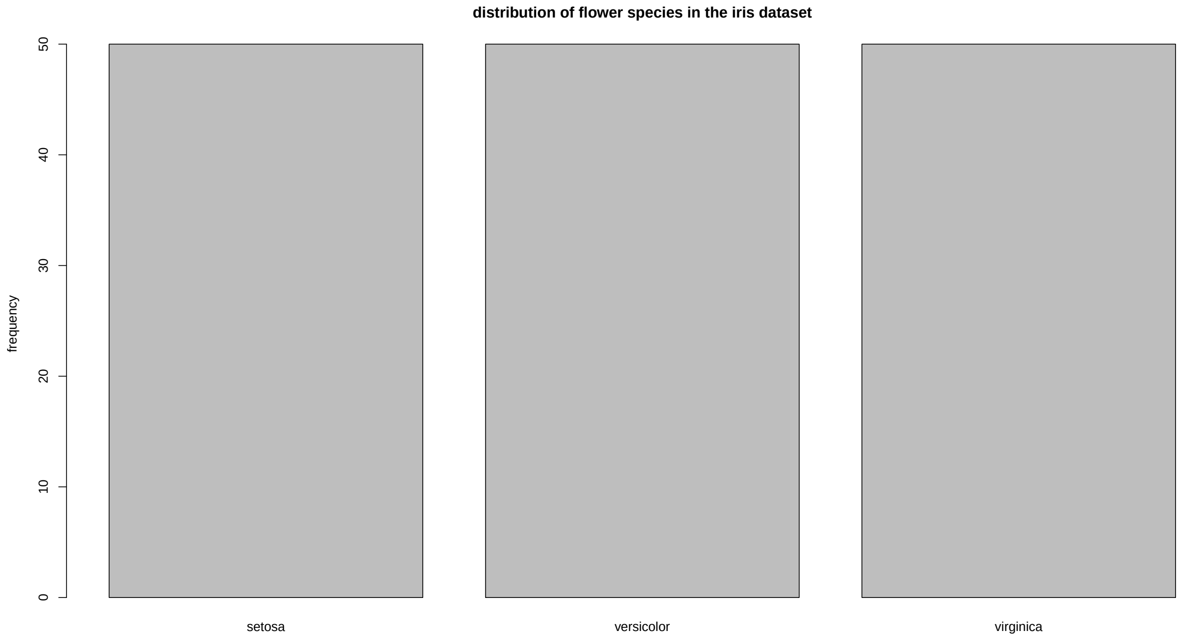
- Histograms using ggplot2:

```
g<-list()
g[[1]]<-ggplot(data=iris,mapping=aes(x=Sepal.Length)) + geom_histogram(bins=25)
g[[2]]<-ggplot(data=iris,mapping=aes(x=Sepal.Width)) + geom_histogram(bins=25)
g[[3]]<-ggplot(data=iris,mapping=aes(x=Petal.Length)) + geom_histogram(bins=25)
g[[4]]<-ggplot(data=iris,mapping=aes(x=Petal.Width)) + geom_histogram(bins=25)
grid.arrange(g[[1]],g[[2]],g[[3]],g[[4]],nrow=2)
```



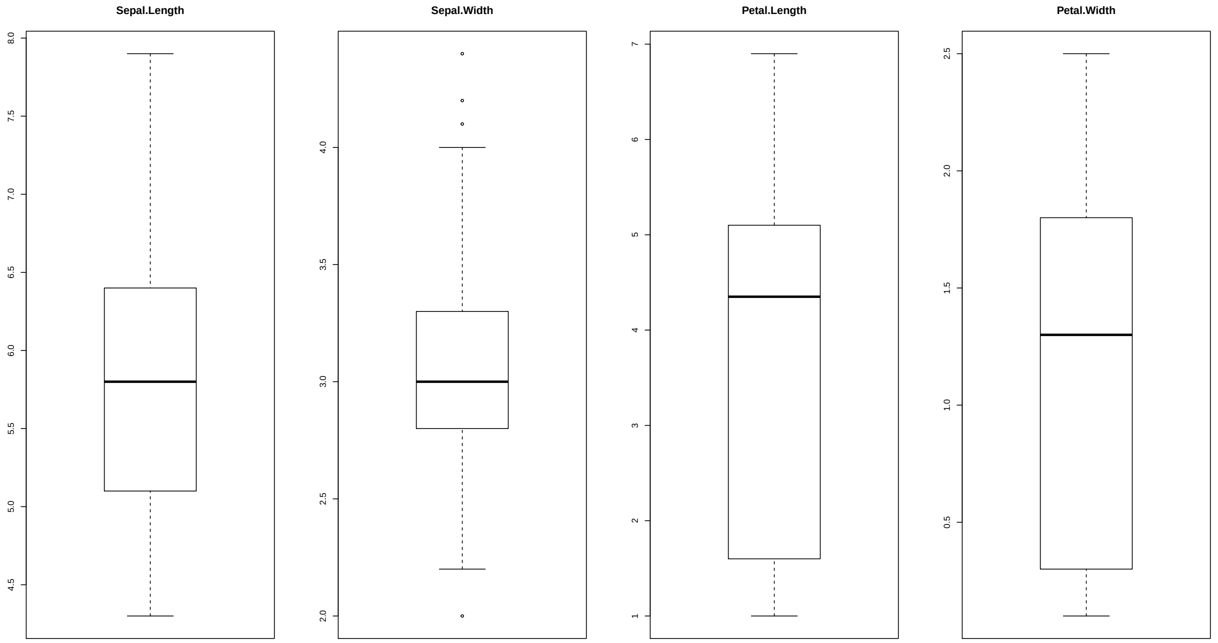
- Barplot

```
barplot(table(iris$Species),
        main="distribution of flower species in the iris dataset",
        ylab="frequency")
```



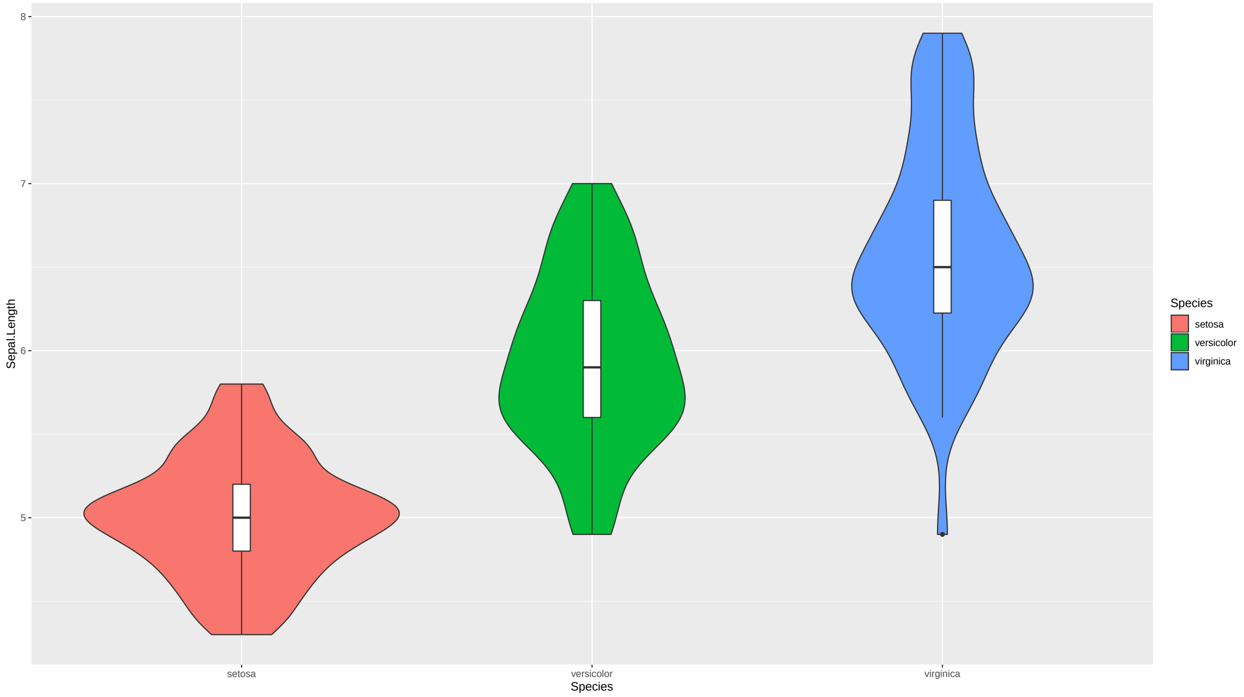
- Box plots

```
par(mfrow=c(1,4))
for(i in 1:4) {
  boxplot(iris[,i], main=names(iris)[i])
}
```



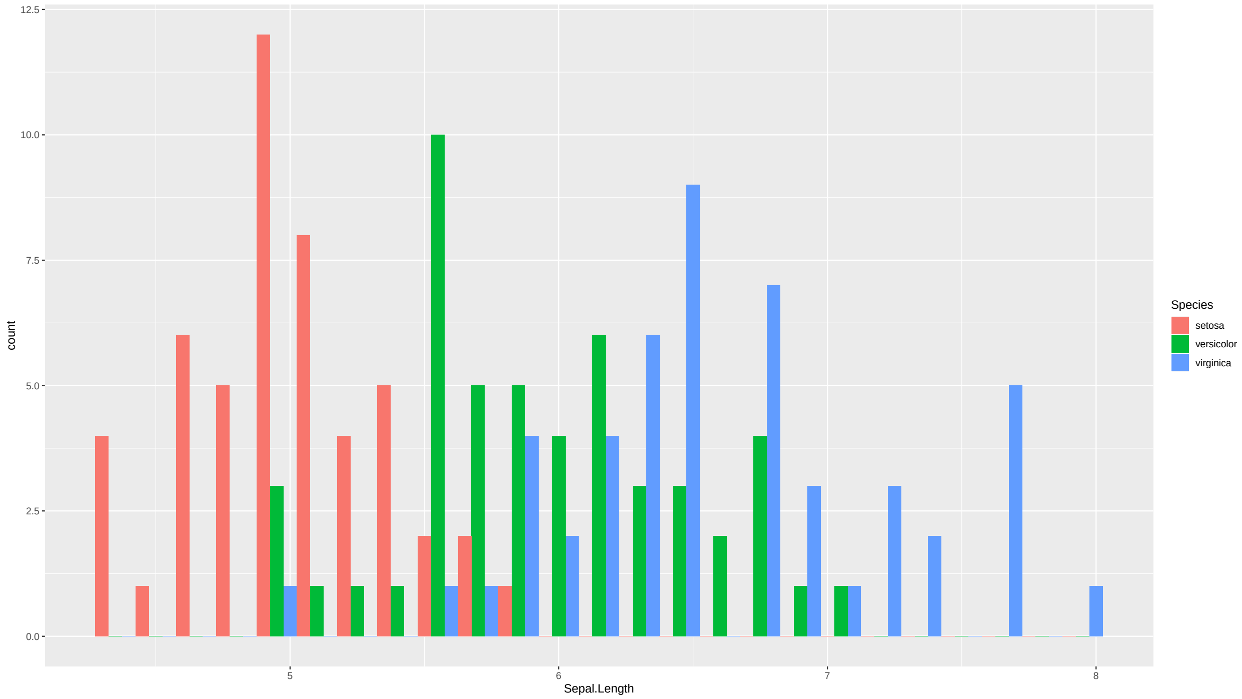
- Violin plot of `Sepal.Length`

```
ggplot(data=iris,mapping=aes(x=Species,y=Sepal.Length,fill=Species)) +
  geom_violin() +
  geom_boxplot(width=0.05, fill="white")
```



- Histograms for `Sepal.Length`

```
ggplot(data=iris,mapping=aes(x=Sepal.Length,fill=Species)) +
  geom_histogram(binwidth=0.15,position="dodge")
```



- Pair-wise scatterplots

```
par(mfrow=c(2,3))
for(i in 1:3){
  for(j in min(c(i+1),4):4){
    print(c(i,j))
    plot(iris[,i],
         iris[,j],
         pch=20,
         col=ifelse(iris$Species=="setosa","salmon",ifelse(iris$Species=="versicolor","darkgreen","steel
         xlab=colnames(iris)[i],
         ylab=colnames(iris)[j])
  }
}
```

```
## [1] 1 2
```

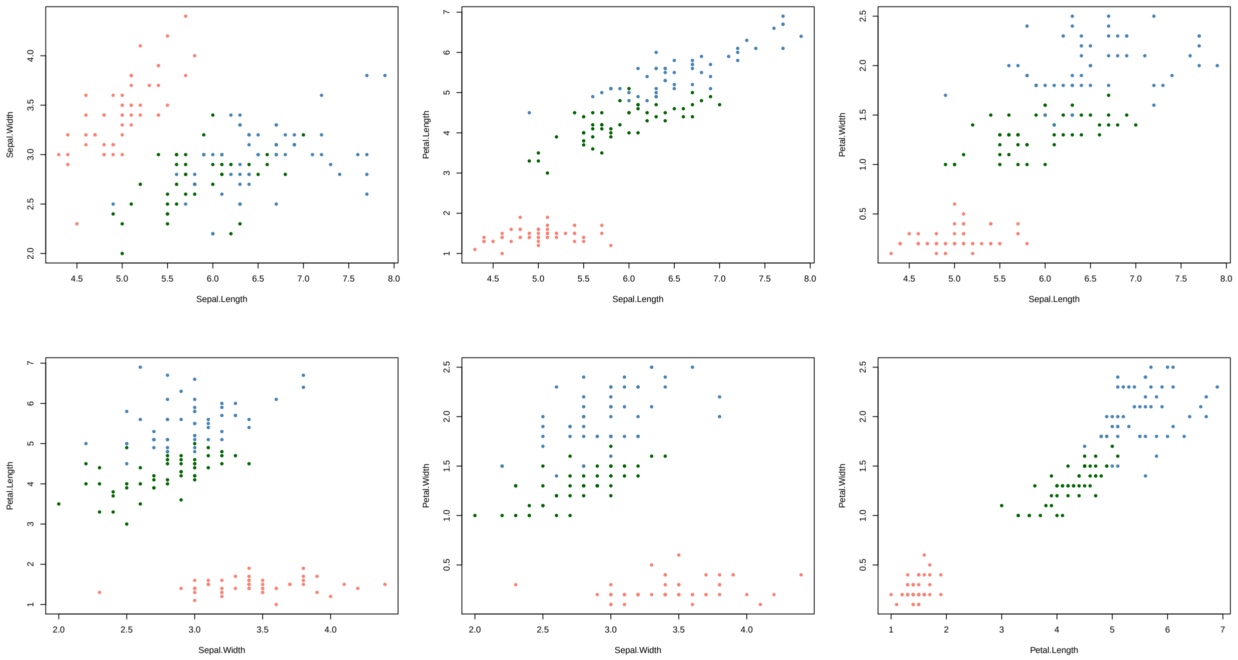
```
## [1] 1 3
```

```
## [1] 1 4
```

```
## [1] 2 3
```

```
## [1] 2 4
```

```
## [1] 3 4
```



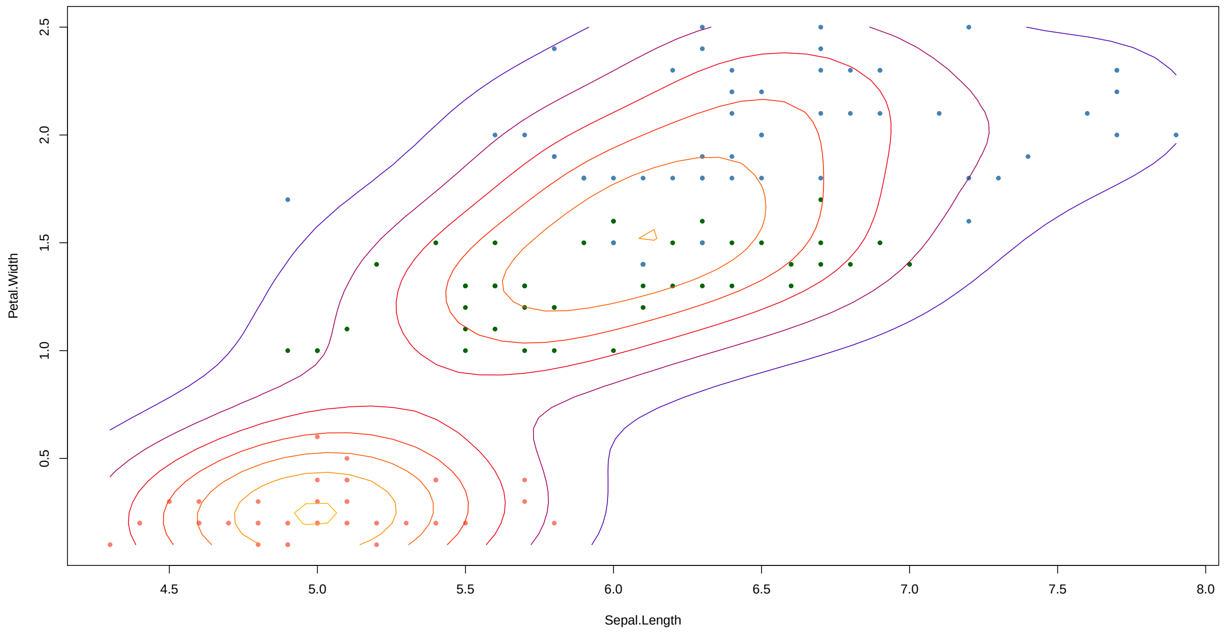
- Bivariate density contours

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
clrs<-colorRampPalette(c("blue","red","orange","yellow","white"))
z <- kde2d(iris$Sepal.Length, iris$Petal.Width, n=50)
plot(iris[,c("Sepal.Length","Petal.Width")],
     xlab="Sepal.Length",
     ylab="Petal.Width",
     col=ifelse(iris$Species=="setosa","salmon",ifelse(iris$Species=="versicolor","darkgreen","steelblu
     pch=20)
contour(z, drawlabels=FALSE, nlevels=11,col=clrs(14), add=TRUE)
```

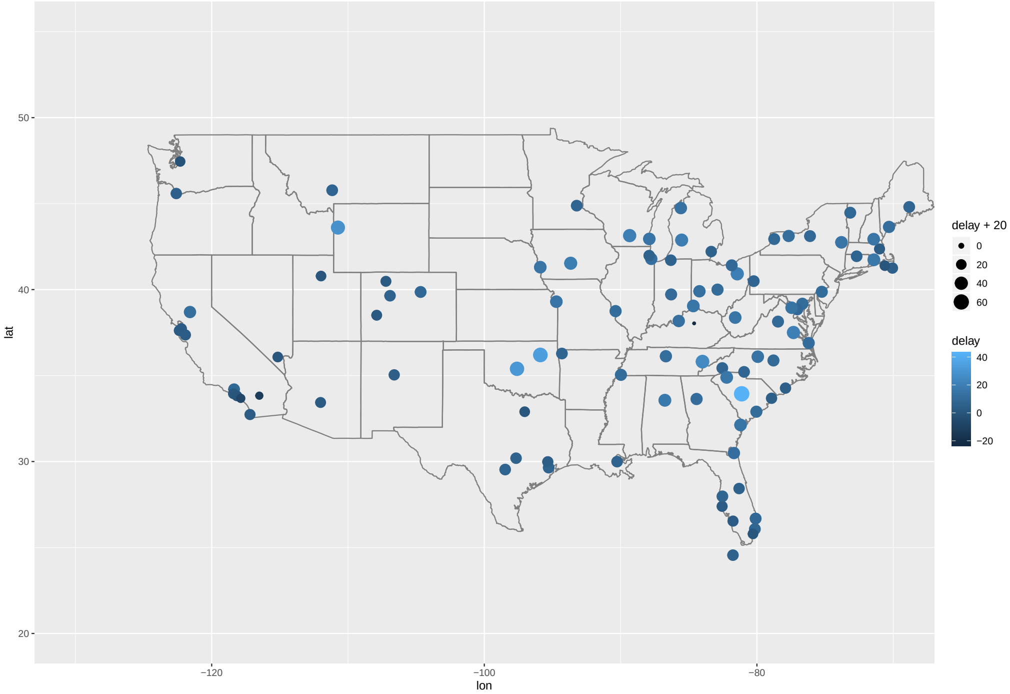


2. Install the package `nycflights13`, then load it. This has data on flights that took off in the US during 2013. There are 5 data tables:
   - `airlines`, data on airlines
   - `airports`, data on airports
   - `planes`, data on planes
   - `weather`, hourly weather data at NYC airports for 2013
   - `flights`, data on flights leaving NYC airports during 2013

- Compute the average delay by destination, then join the airports data frame to get the longitude and latitude of delays. Plot this (if you are using ggplot2, then the functions `borders()` and `coord_quickmap()` can be useful for a nicer figure).
- Construct data frames giving average delay per wind speed / temperature / precipitation / visibility. Produce scatter plots of each of these against delay and add an average trend line.

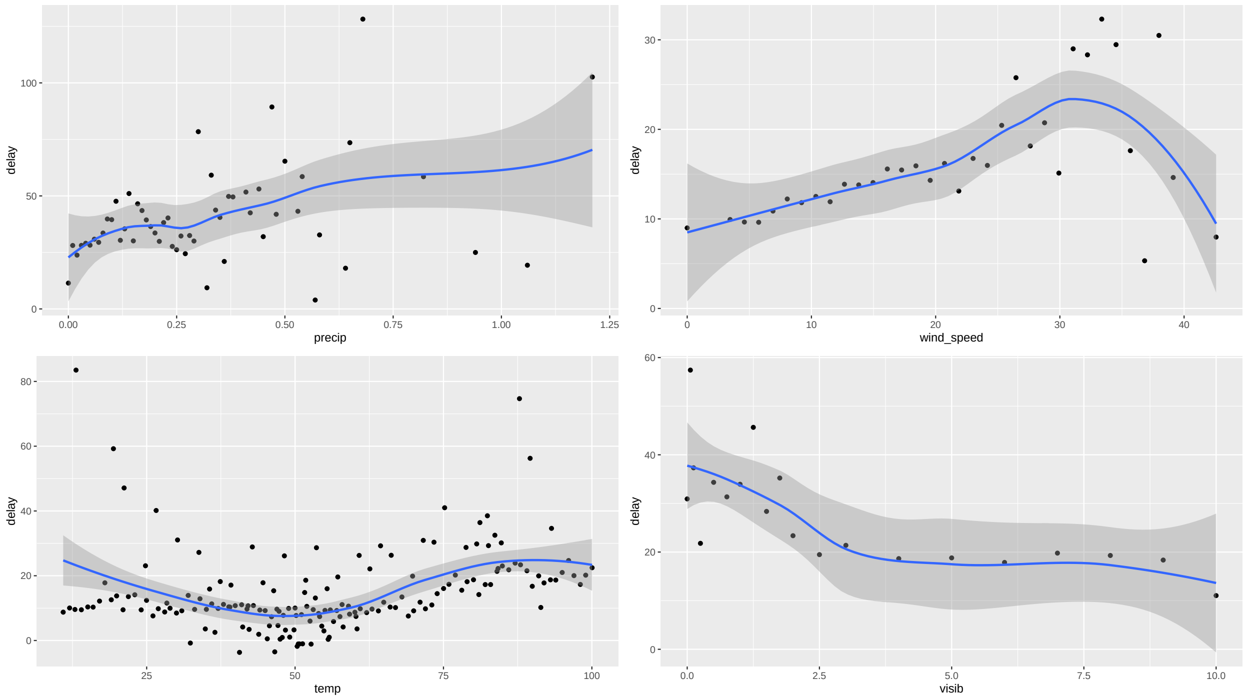
```
library(nycflights13)

# Compute the average delay by destination, then join the airports data frame
# to get the longitude and latitude of delays.
avg_dest_delays <-
    flights %>%
    group_by(dest) %>%
    summarise(delay = mean(arr_delay, na.rm = TRUE)) %>% # arrival delay NA's are cancelled flights
    inner_join(airports, by = c(dest = "faa"))

# stratify by origin airport
avg_dest_delays_by_origin <-
    flights %>%
    group_by(origin,dest) %>%
    summarise(delay = mean(arr_delay, na.rm = TRUE)) %>% # arrival delay NA's are cancelled flights
    inner_join(airports, by = c(dest = "faa"))

# plotting this
```

```
ggplot(data=avg_dest_delays,mapping=aes(lon,lat,colour=delay,size=delay+20)) +
  borders("state") +
  geom_point() +
  coord_quickmap(xlim=c(-130,-70),ylim=c(20,55)) # xlim, ylim to hide Alaska and Hawaii
```



```
flights_weather <-flights %>% left_join(weather,by=c("year","month","day","hour"))

flights_precip <- flights_weather %>%
  group_by(precip) %>%
  summarise(delay=mean(dep_delay,na.rm=T))

flights_wind <- flights_weather %>%
  group_by(wind_speed) %>%
  summarise(delay=mean(dep_delay,na.rm=T))

flights_temp <- flights_weather %>%
  group_by(temp) %>%
  summarise(delay=mean(dep_delay,na.rm=T))

flights_visib <- flights_weather %>%
  group_by(visib) %>%
  summarise(delay=mean(dep_delay,na.rm=T))

g1<-ggplot(data=flights_precip,mapping=aes(x=precip,y=delay)) +
  geom_point() +
  geom_smooth()

g2<-ggplot(data=flights_wind,mapping=aes(x=wind_speed,y=delay)) +
  geom_point() +
  geom_smooth()
```

```r
g3<-ggplot(data=flights_temp,mapping=aes(x=temp,y=delay)) +
  geom_point() +
  geom_smooth()

g4<-ggplot(data=flights_visib,mapping=aes(x=visib,y=delay)) +
  geom_point() +
  geom_smooth()

grid.arrange(g1,g2,g3,g4)
```


delay
precip
delay
wind_speed
delay
temp
delay
visib