MLW / KUHeS Statistics and R short course

Session 2 - Practical (solutions)

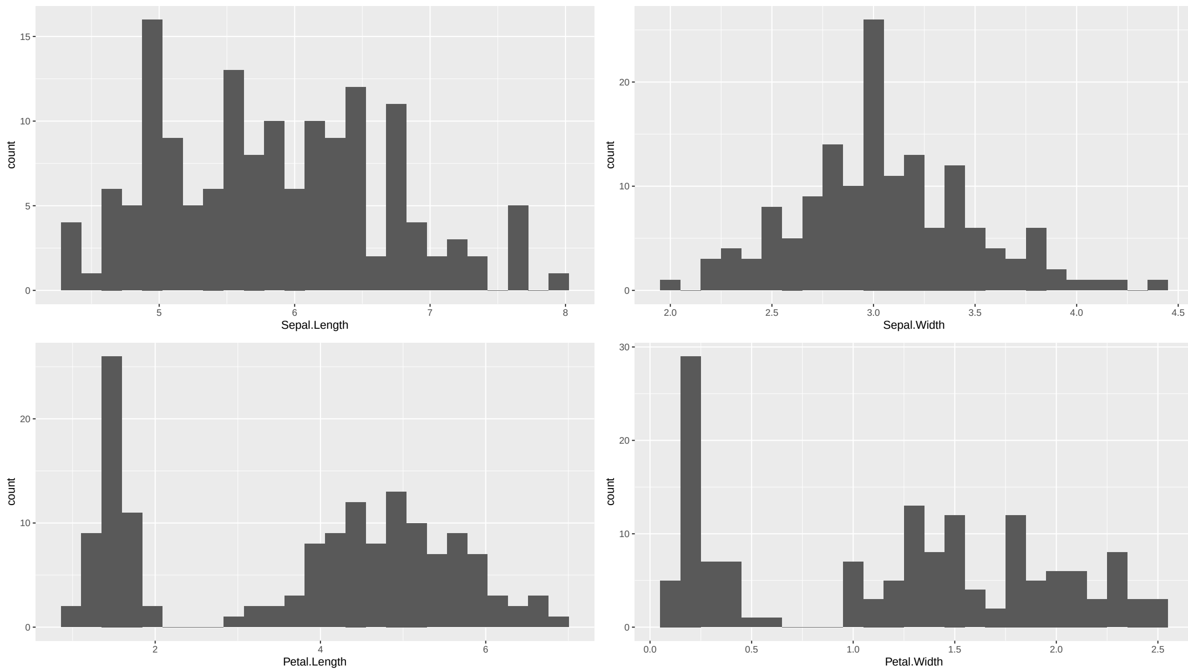
Marc Henrion, Evaristar Kudowa, Vester Gunsaru

2024-10-09

Session 2 - Practical (Solutions)

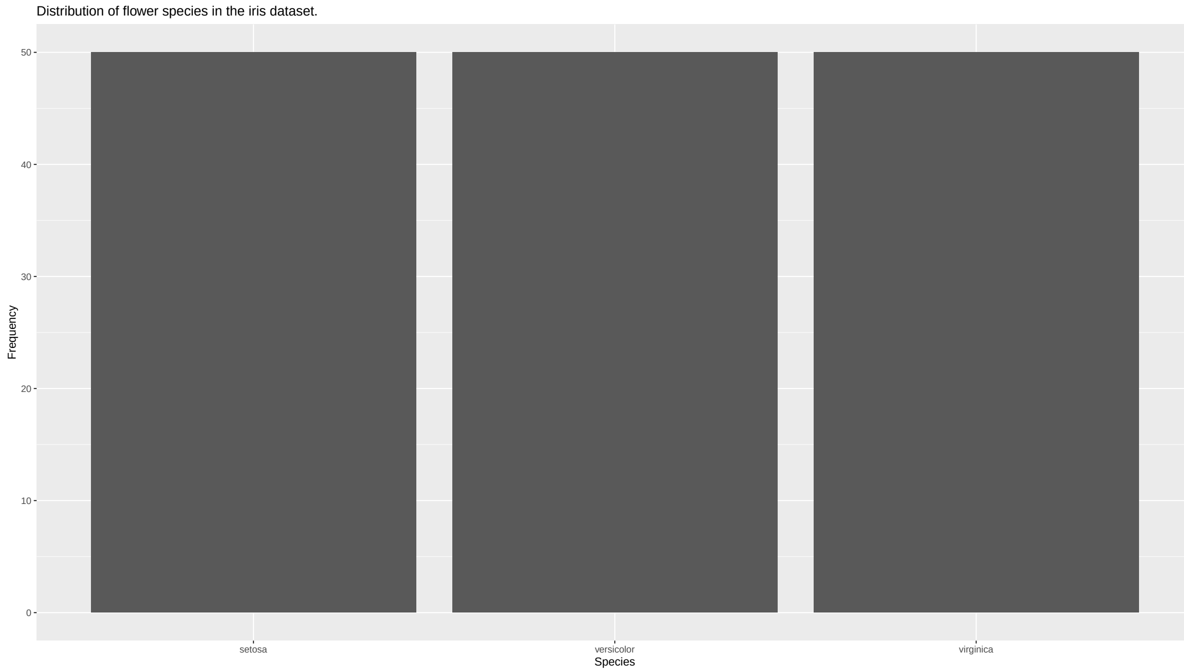
1. Using the `iris` dataset (type `?iris` to get more information about this dataset) that comes pre-loaded with R, produce the following figures:
    - Produce histograms for each of `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`.
    - Produce a bar plot for `Species`.
    - Produce box 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.

```r
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

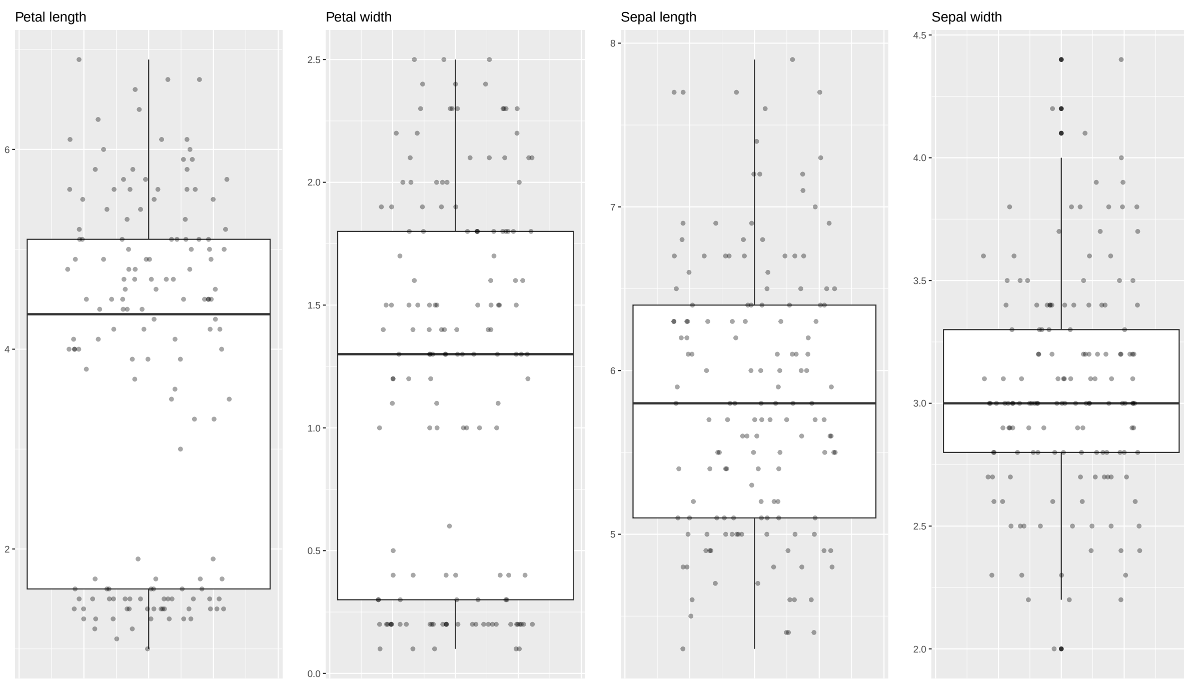
```
iris %>%
  ggplot(mapping=aes(x=Species)) +
  geom_bar() +
  labs(title="Distribution of flower species in the iris dataset.") +
  ylab("Frequency")
```



- Box plots

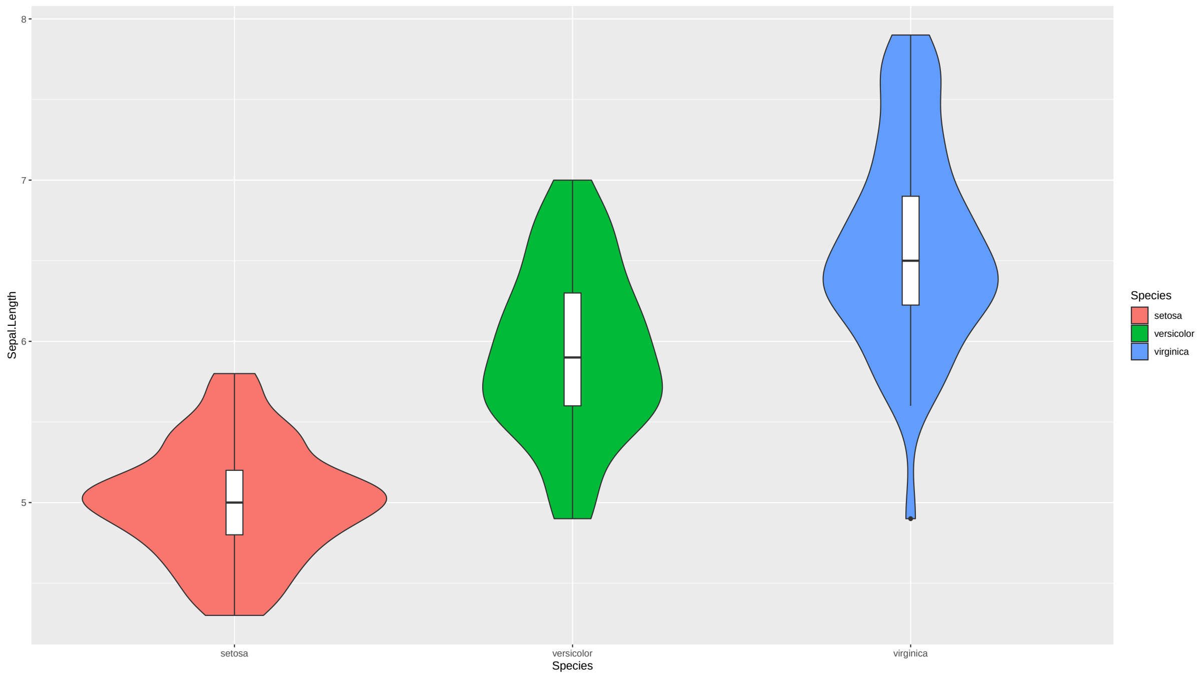
```
g1<-iris %>%
  ggplot(mapping=aes(x=1,y=Petal.Length)) +
  geom_boxplot() +
  geom_jitter(height=0,width=0.25,alpha=0.35) +
  labs(title="Petal length") +
  ylab("") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

g2<-iris %>%
  ggplot(mapping=aes(x=1,y=Petal.Width)) +
  geom_boxplot() +
  geom_jitter(height=0,width=0.25,alpha=0.35) +
  labs(title="Petal width") +
  ylab("") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

g3<-iris %>%
  ggplot(mapping=aes(x=1,y=Sepal.Length)) +
  geom_boxplot() +
  geom_jitter(height=0,width=0.25,alpha=0.35) +
```

```
  labs(title="Sepal length") +
  ylab("") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

g4<-iris %>%
  ggplot(mapping=aes(x=1,y=Sepal.Width)) +
  geom_boxplot() +
  geom_jitter(height=0,width=0.25,alpha=0.35) +
  labs(title="Sepal width") +
  ylab("") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

grid.arrange(g1,g2,g3,g4,nrow=1)
```



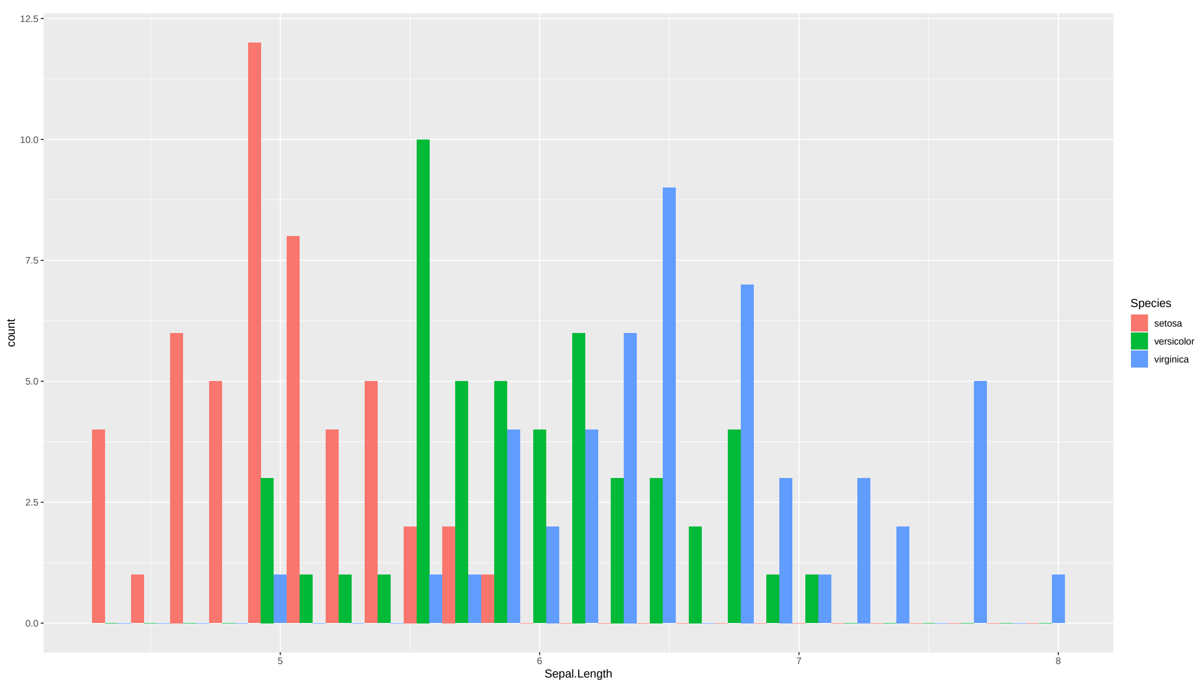
- Violin plot of `Sepal.Length`

```
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

```
g<-list()
counter<-0

for(i in 1:3){
  for(j in min(c(i+1),4):4){
    counter<-counter+1

    g[[counter]]<-iris %>%
      ggplot(mapping=aes(x=get(colnames(iris)[i]),y=get(colnames(iris[j])),col=Species)) +
      geom_point() +
      scale_color_manual(values=c("steelblue","orange","salmon")) +
      xlab(colnames(iris)[i]) +
      ylab(colnames(iris)[j])
  }
}
```

- Bivariate density contours

This requires a bit of extra work and so you have likely found this harder:

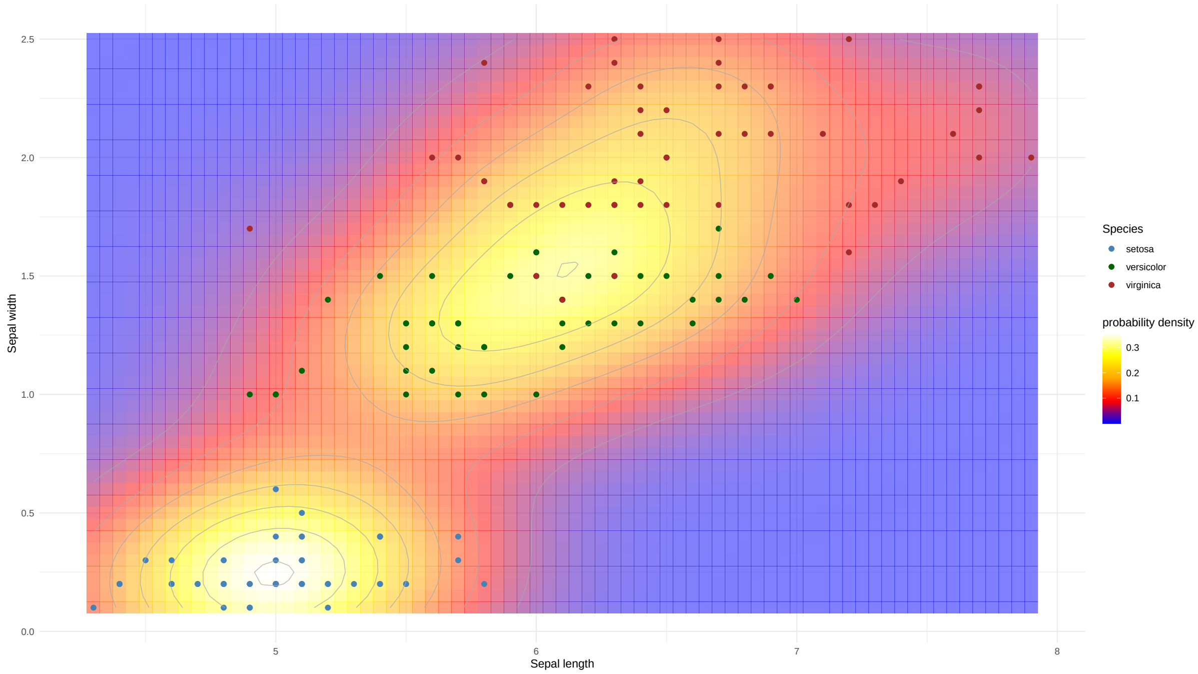
1. Estimate the 2-dimensional density.
2. Using multiple geoms with different datasets.

```
library(MASS)
```

```

Attaching package: 'MASS'

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

    select
```

```
clrs<-colorRampPalette(c("blue","red","orange","yellow","white"))
dens <- kde2d(iris$Sepal.Length,
              iris$Petal.Width,
              n=c(length(seq(min(iris$Sepal.Length),max(iris$Sepal.Length),by=0.05)),
                  length(seq(min(iris$Petal.Width),max(iris$Petal.Width),by=0.05))))

df<-expand.grid(dens$x,dens$y)
df$z<-as.vector(dens$z)
colnames(df)<-c("x","y","z")

ggplot() +
  geom_tile(data=df,mapping=aes(x=x,y=y,fill=z,z=z),width=0.05,height=0.05,alpha=0.5) +
```

```
  geom_point(data=iris,mapping=aes(x=Sepal.Length,y=Petal.Width,col=Species),size=2) +
  geom_contour(data=df,mapping=aes(x=x,y=y,fill=z,z=z),col="darkgrey",lwd=0.35,alpha=0.75) +
  scale_fill_gradientn(colours = clrs(200),name="probability density") +
  scale_color_manual(values=c("steelblue","darkgreen","brown")) +
  theme_minimal() +
  xlab("Sepal length") +
  ylab("Sepal width")
```



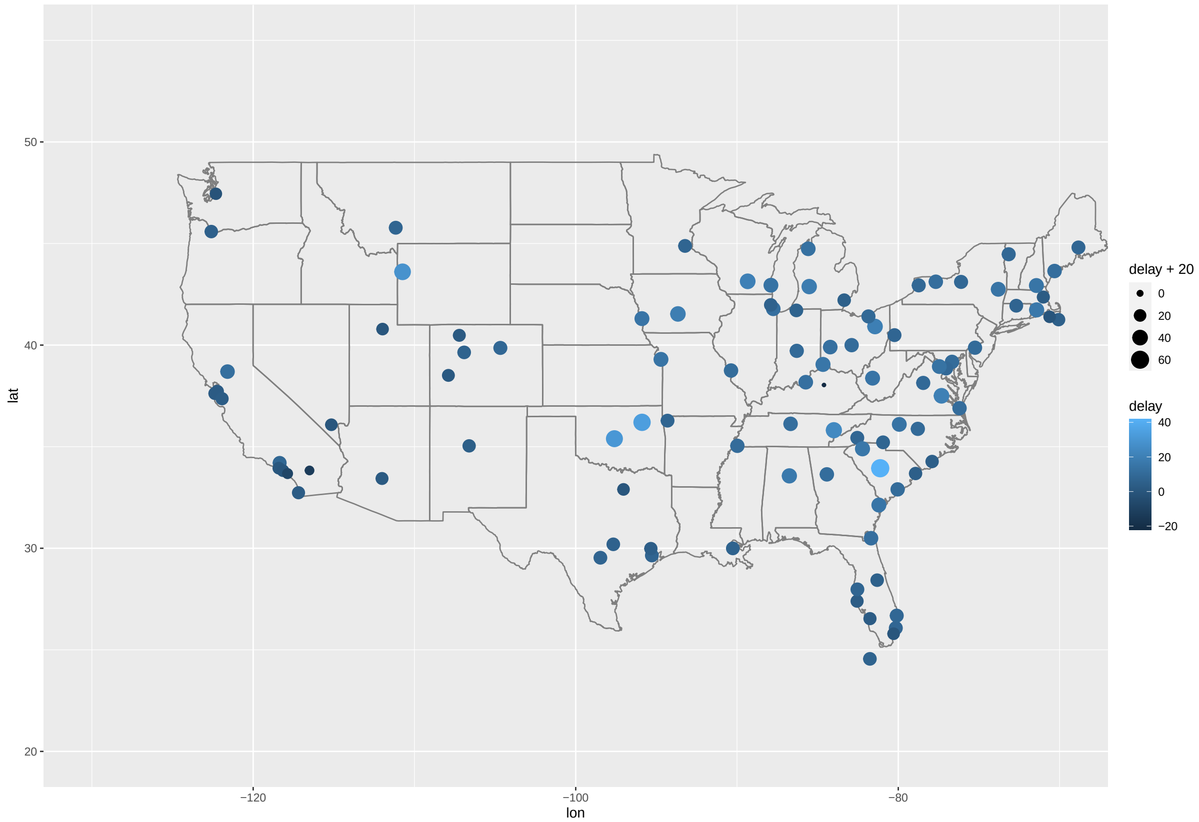
2. Install the package `nycflights13`, then load it. This has data on flights that took off in the US during 2013. There are 5 data tables:

   - `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

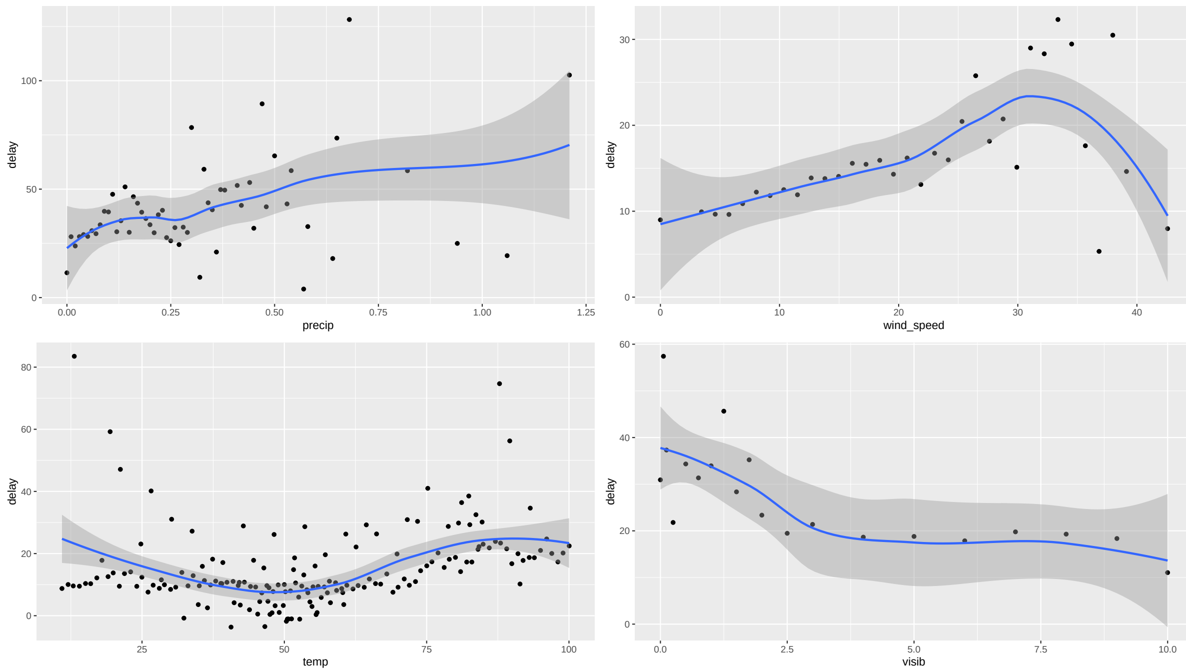
- Compute the average delay by destination, then join the airports data frame to get the longitude and latitude of delays. Plot this (if you are using ggplot2, then the functions `borders()` and `coord_quickmap()` can be useful for a nicer figure).
- Construct data frames giving average delay per wind 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
```



```r
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()

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