Lecture 15. Going Further with the Grammar of Graphics

R and Data Visualization

BIG2006, Hanyang University, Fall 2022

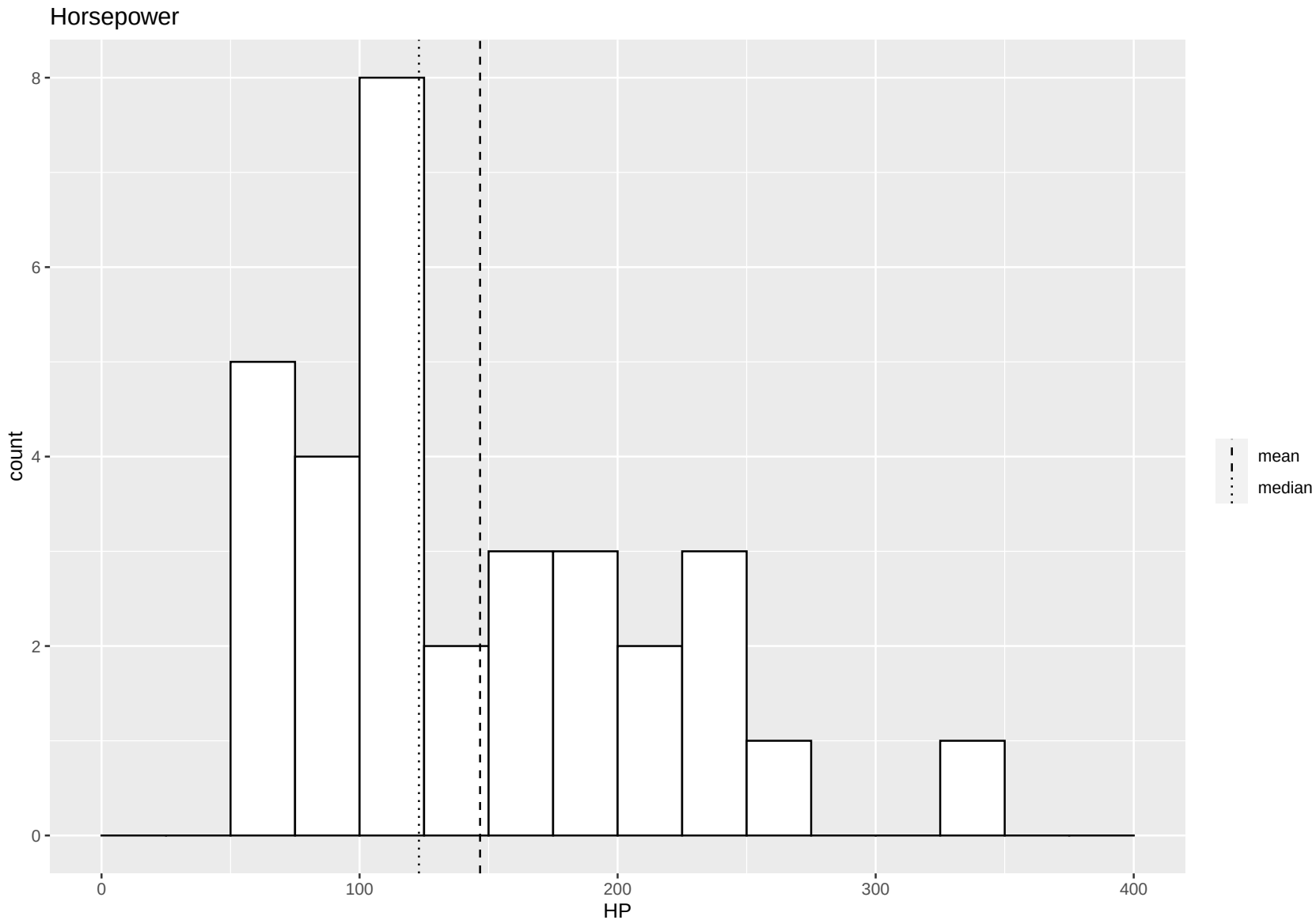
ggplot or qplot?

There are several key differences between ggplot and qplot:

- qplot is a shortcut version of ggplot.
- ggplot prefers its data argument as a data frame object, and you tell it what to do by explicitly adding geom layers.
- A call to qplot alone can produce a graphic. When using ggplot, layers have to be added before anything becomes visible.
- To access the full power and flexibility of ggplot2 graphics, ggplot is the recommended function and this comes at the cost of providing a little more explicit instruction.

- `gg.static`: represents the part of the plot that will stay the same throughout if you want to experiment with adding other features later.
- The addition of the `geom_histogram` layer to `gg.static` invokes the plot, and the addition of `gg.lines` with changes to the default line types made with `scale_linetype_manual` marks off the mean and median.

```
gg.static <- ggplot(data=mtcars,mapping=aes(x=hp)) +
  ggtitle("Horsepower") + labs(x="HP")
mtcars.mm <- data.frame(mm=c(mean(mtcars$hp),median(mtcars$hp)),
                        stats=factor(c("mean","median")))
gg.lines <- geom_vline(mapping=aes(xintercept=mm,linetype=stats),
                       show.legend=TRUE,data=mtcars.mm)
gg.static + geom_histogram(color="black",fill="white",
                           breaks=seq(0,400,25),closed="right") +
  gg.lines + scale_linetype_manual(values=c(2,3)) + labs(linetype="")
```


Horsepower
count
HP
0
2
4
6
8
0
100
200
300
400
mean
median

Smoothing and Shading

Data visualization using the `ggplot2` package is particularly powerful when you want to split features of the plot by one or more categorical variables.

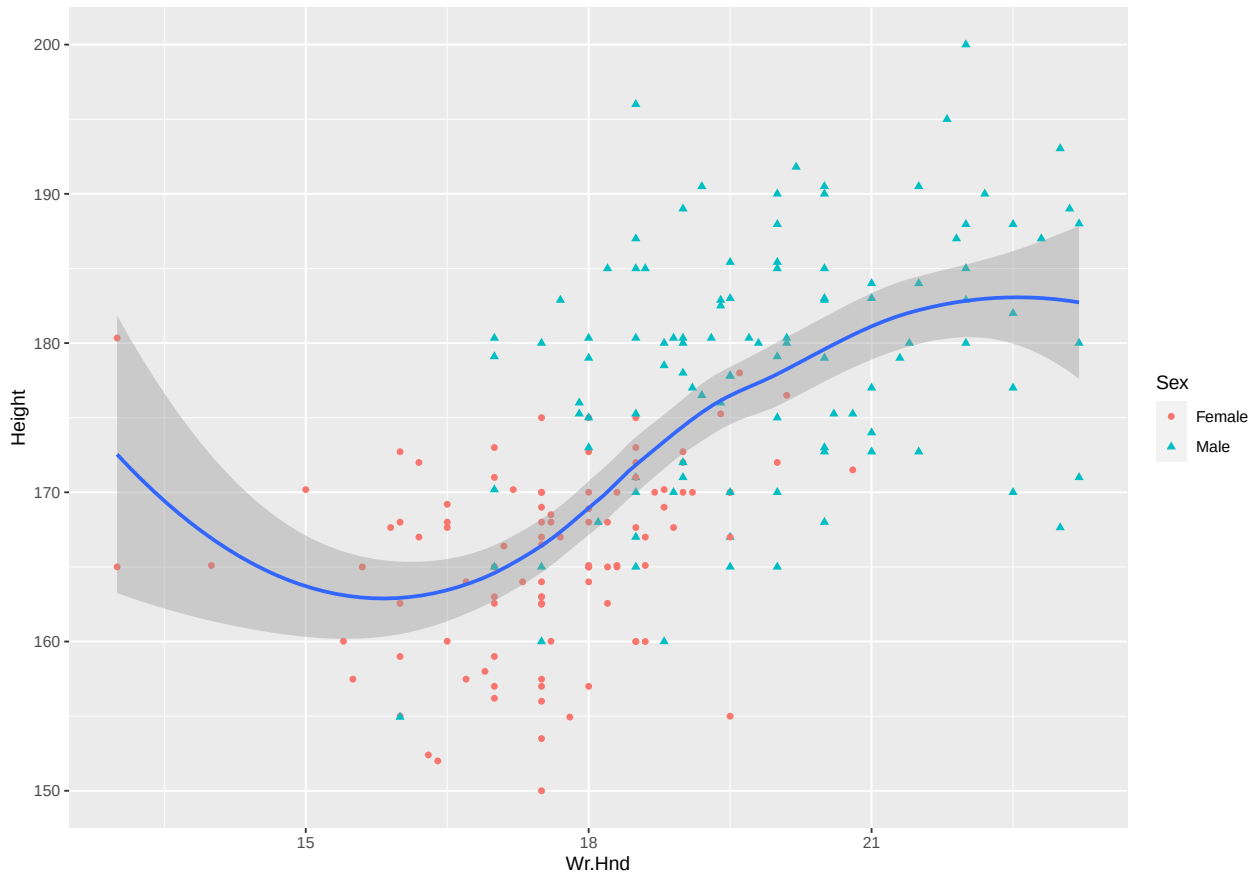
Adding LOESS Trends

- Nonparametric smoothing: determines how your data appear to behave without fitting a specific model
- Flexible aids for interpreting over all trends, whatever their form
- Locally weighted scatterplot smoothing (LOESS or LOWESS): produces the smoothed trend by using regression methods on localized subsets of the data, step-by-step over the entire range of the explanatory variable.

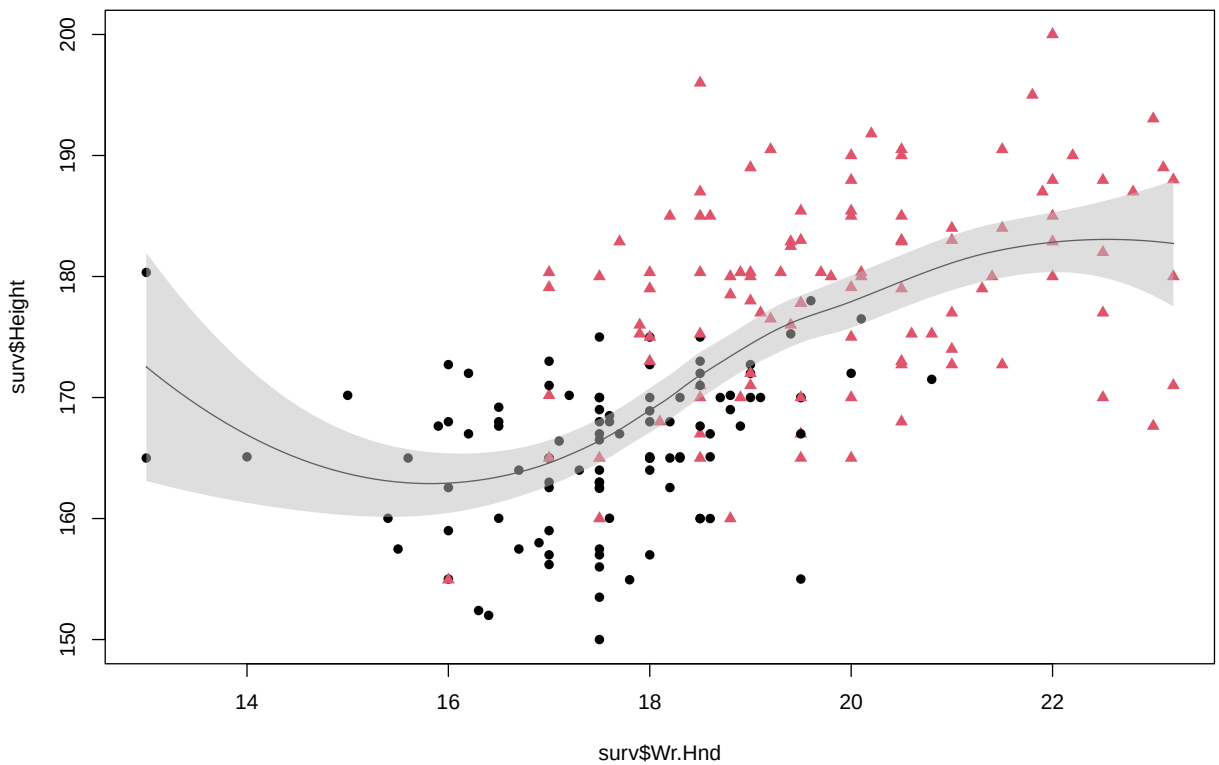
Note: The trade-off is that you are not provided with any specific details of the relationship between response and predictors and you lose any reliable ability to extrapolate.

```r
surv <- na.omit(survey[,c("Sex","Wr.Hnd","Height")])
ggplot(surv,aes(x=Wr.Hnd,y=Height)) +
  geom_point(aes(col=Sex,shape=Sex)) + geom_smooth(method="loess")
```

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

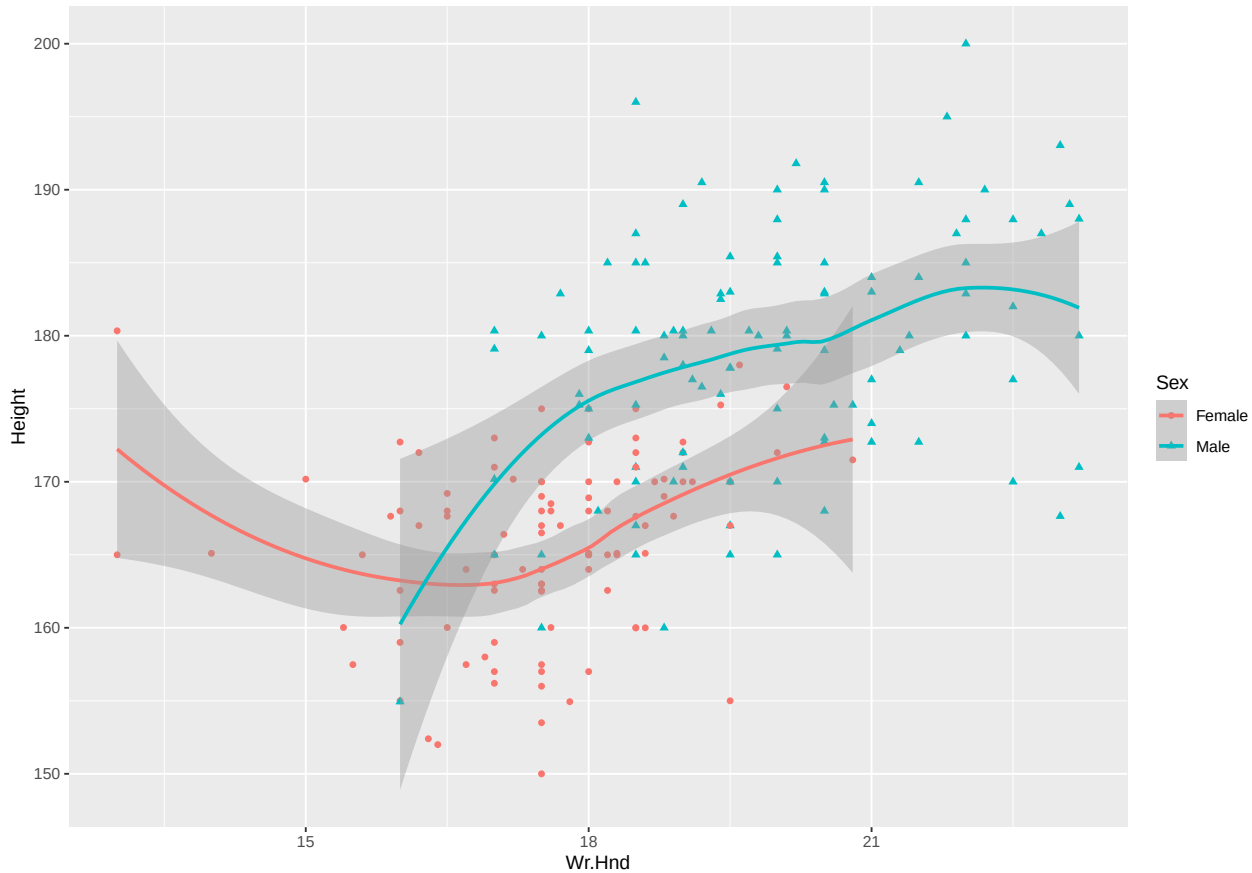


```
plot(surv$Wr.Hnd,surv$Height,col=surv$Sex,pch=c(16,17)[surv$Sex])
smoother <- loess(Height~Wr.Hnd,data=surv)
handseq <- seq(min(surv$Wr.Hnd),max(surv$Wr.Hnd),length=100)
sm <- predict(smoother,newdata=data.frame(Wr.Hnd=handseq),se=TRUE)
lines(handseq,sm$fit)
polygon(x=c(handseq,rev(handseq)),y=c(sm$fit+2*sm$se,rev(sm$fit-2*sm$se)),
        col=adjustcolor("gray",alpha.f=0.5),border=NA)
```



```
ggplot(surv,aes(x=Wr.Hnd,y=Height,col=Sex,shape=Sex)) +
  geom_point() + geom_smooth(method="loess")
```

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



Note: The implementation of LOESS and other trend smoothers depends on your specifying the amount of smoothing you want. This is controlled by the proportion of the data to use as each localized weighted subset, for each step/location in the estimation procedure.

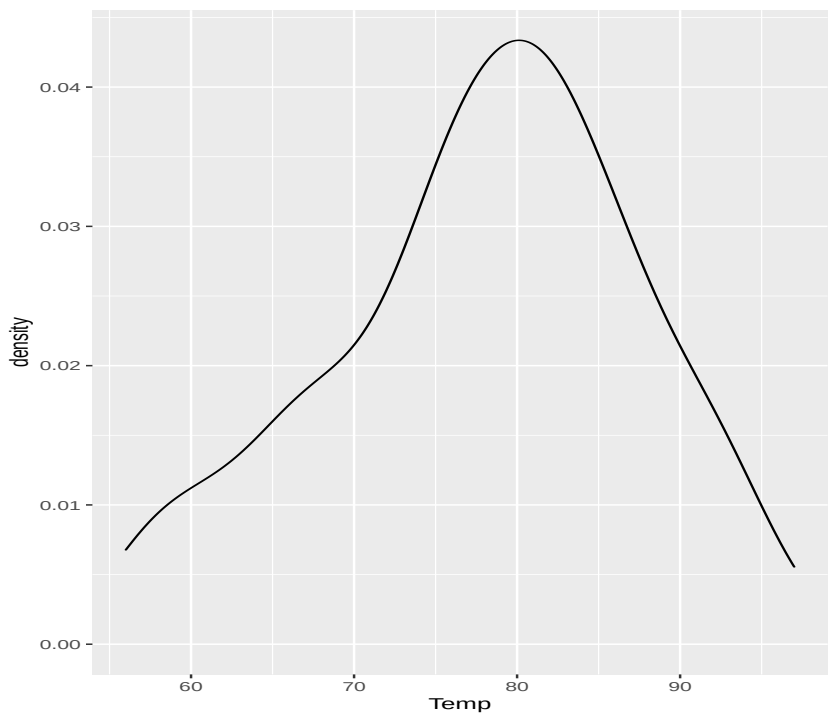
A larger proportion leads to a smoother, less variable trend estimate than a smaller proportion. This value, referred to as the $span$, can be set by the optoinal argument `span` in either `loess` or `geom_smooth`.

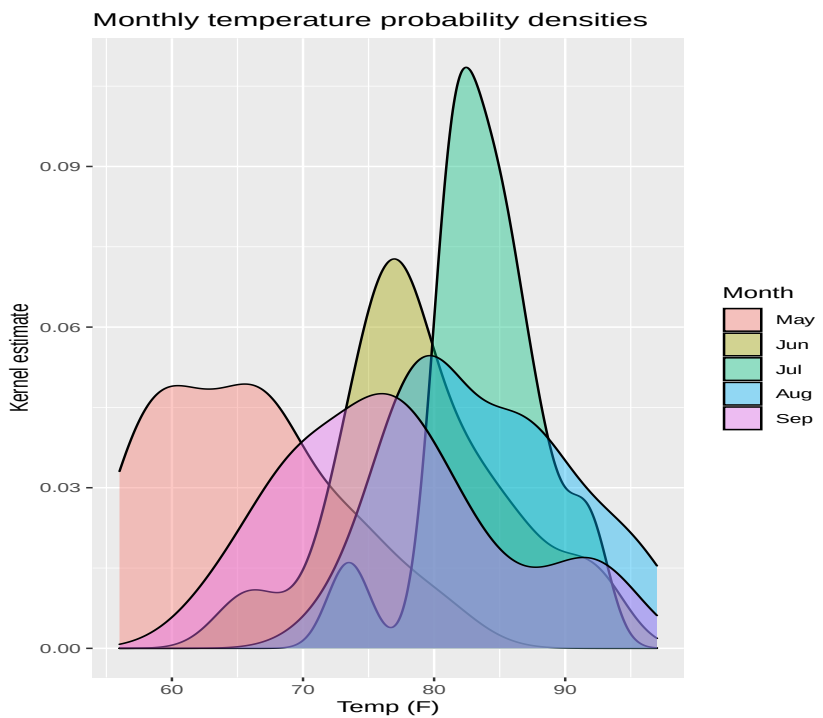
Smooth Density Estimates

Kernel density estimation (KDE) is a method for producing a smooth estimate of a probability density function, based on observed data.

KDE involves assigning a scaled probability function (the *kernel*) to each observation in a data set and summing them all to give an impression of the distribution of the data set as a whole.

```r
a <- ggplot(data=airquality,aes(x=Temp)) + geom_density()
air <- airquality
air$Month <- factor(air$Month,labels=c("May","Jun","Jul","Aug","Sep"))
b <- ggplot(data=air,aes(x=Temp,fill=Month)) + geom_density(alpha=0.4) +
  ggtitle("Monthly temperature probability densities") +
  labs(x="Temp (F)",y="Kernel estimate")
grid.arrange(a,b, nrow=1, ncol=2) # gridExtra package
```





Note: The precise appearance of kernel-estimated pdf is dependent on the amount of smoothing employed. The quantity of interest in KDE is referred to as the bandwidth or smoothing parameter (a larger bandwidth impose greater smoothing over the range of the data).

By default, the bandwidth is automatically chosen using a data-driven technique and it is generally acceptable for simple exploration of the data.

Multiple Plots and Variable-Mapped Facets

`mfrow` in a call to `par` or compartmentalizing the device using `layout` cannot be used for `ggplot2` graphics.

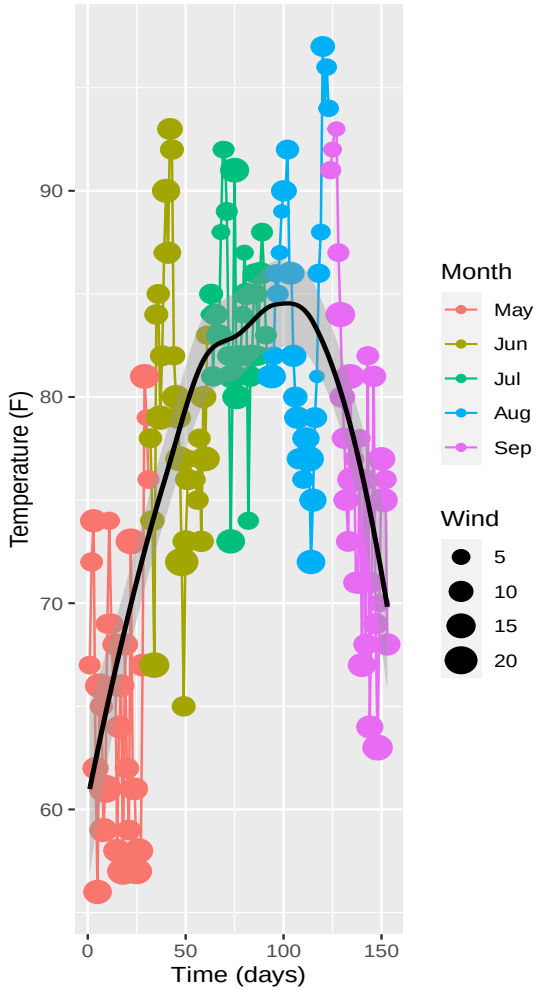
There are other functions that allow independent `ggplot2` plots to populate a single device.

`ggplot2` offers a convenient way to consider multiple-plot graphics using facets.

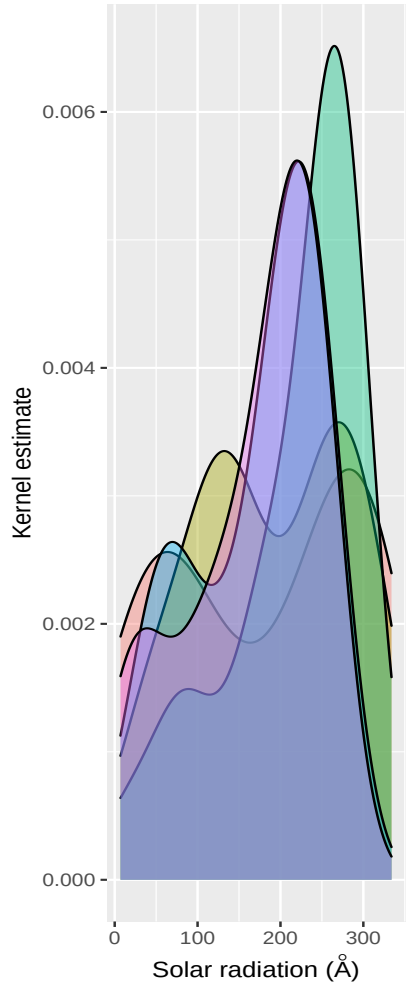
Independent Plots

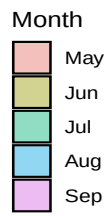
- `grid.arrange` in `gridExtra` package: arranges independent plots as a single image

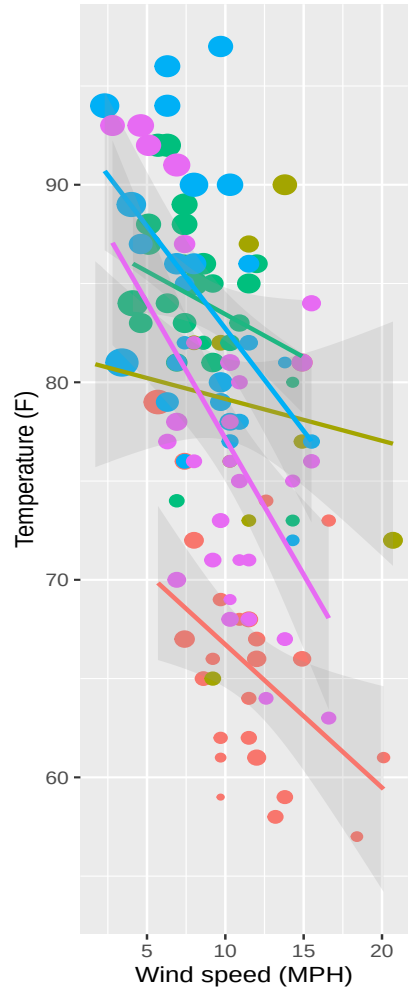
```
gg1 <- ggplot(air,aes(x=1:nrow(air),y=Temp)) + geom_line(aes(col=Month)) +
  geom_point(aes(col=Month,size=Wind)) +
  geom_smooth(method="loess",col="black") +
  labs(x="Time (days)",y="Temperature (F)")
gg2 <- ggplot(air,aes(x=Solar.R,fill=Month)) + geom_density(alpha=0.4) +
  labs(x=expression(paste("Solar radiation (",ring(A),")")),
       y="Kernel estimate")
gg3 <- ggplot(air,aes(x=Wind,y=Temp,color=Month)) +
  geom_point(aes(size=Ozone)) +
  geom_smooth(method="lm",level=0.9,fullrange=FALSE,alpha=0.2) +
  labs(x="Wind speed (MPH)",y="Temperature (F)")
grid.arrange(gg1,gg2,gg3,nrow=1,ncol=3)
```


Temperature (F)
90
80
70
60
0
50
100
150
Time (days)

Month
May
Jun
Jul
Aug
Sep
Wind
5
10
15
20


Kernel estimate
0.006
0.004
0.002
0.000
0
100
200
300
Solar radiation (Å)


Month
May
Jun
Jul
Aug
Sep

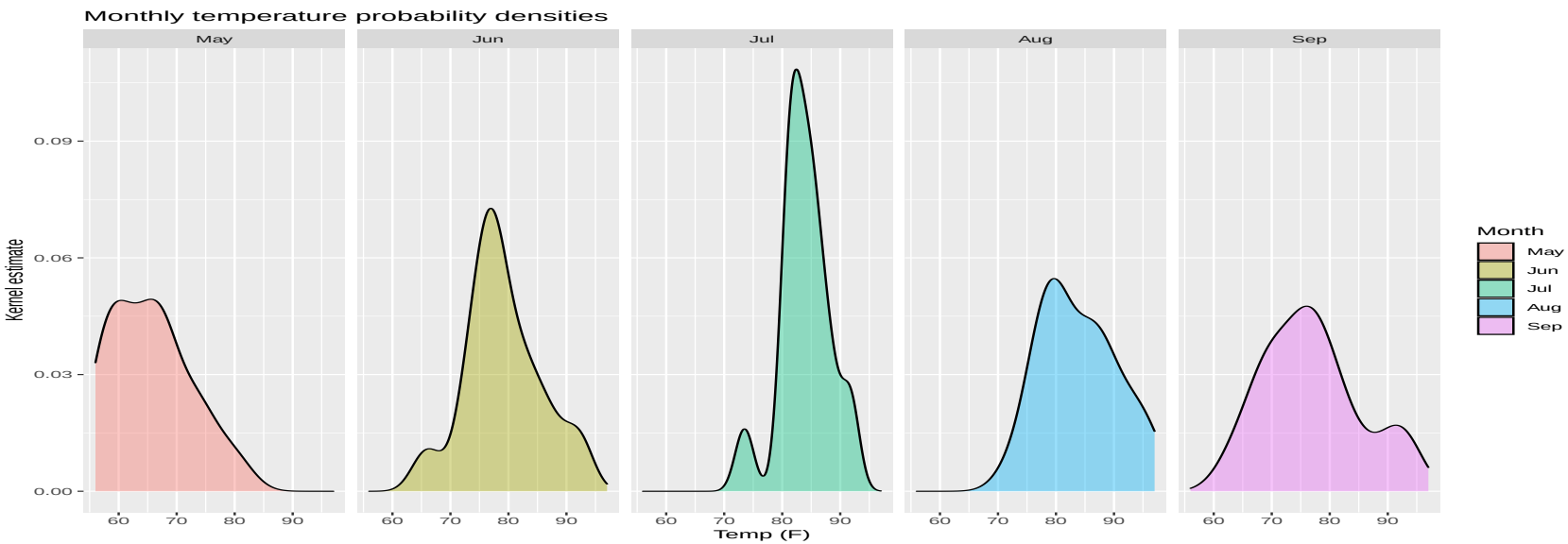

Temperature (F)
90
80
70
60
5
10
15
20
Wind speed (MPH)


Month
May
Jun
Jul
Aug
Sep
Ozone
40
80
120
160

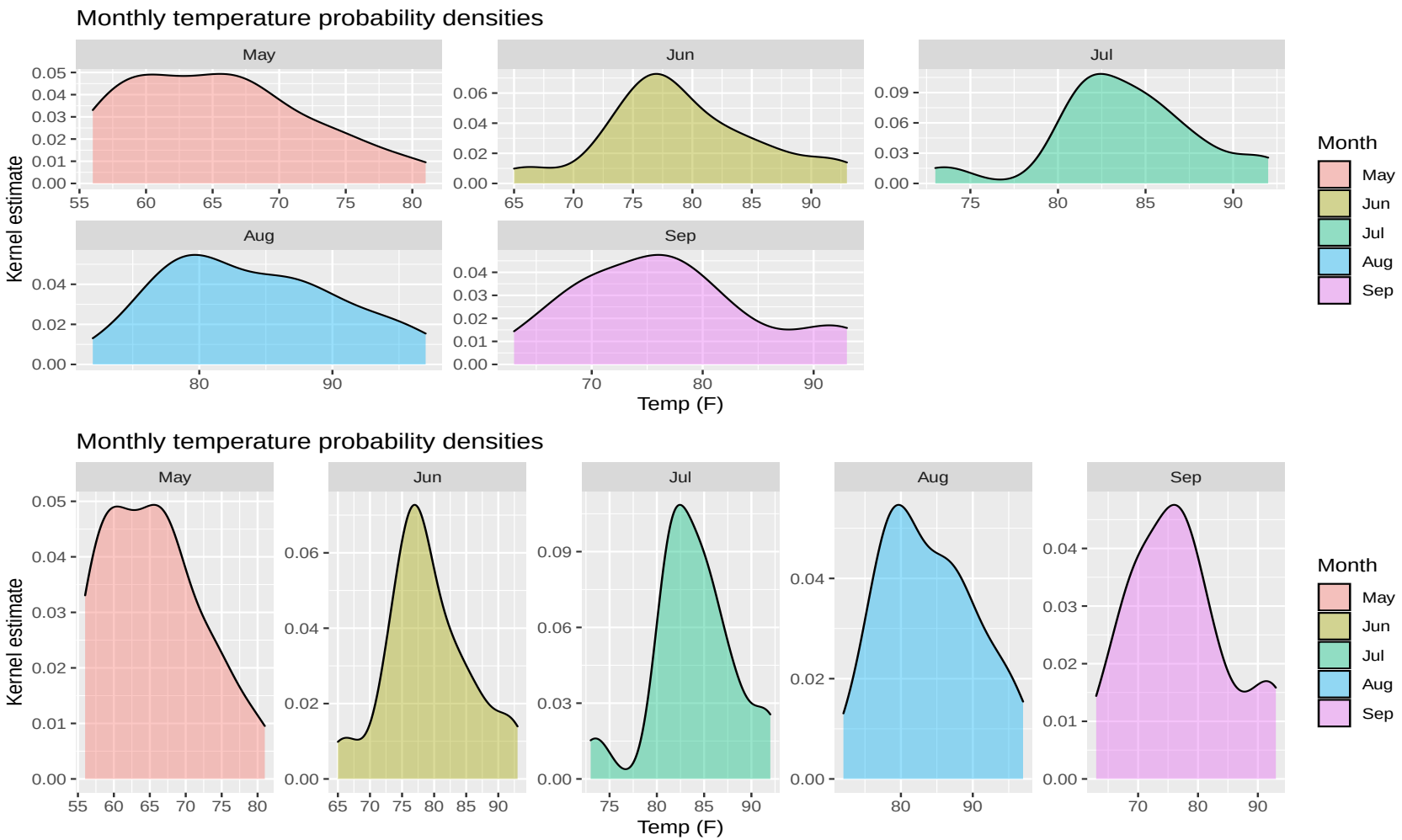
Facets Mapped to a Categorical Variable

- A flexible alternative of `grid.arrange` to view multiple plots

```
ggp <- ggplot(data=air,aes(x=Temp,fill=Month)) + geom_density(alpha=0.4) +
  ggtitle("Monthly temperature probability densities") +
  labs(x="Temp (F)",y="Kernel estimate")
ggp + facet_wrap(~Month,nrow=1)
```



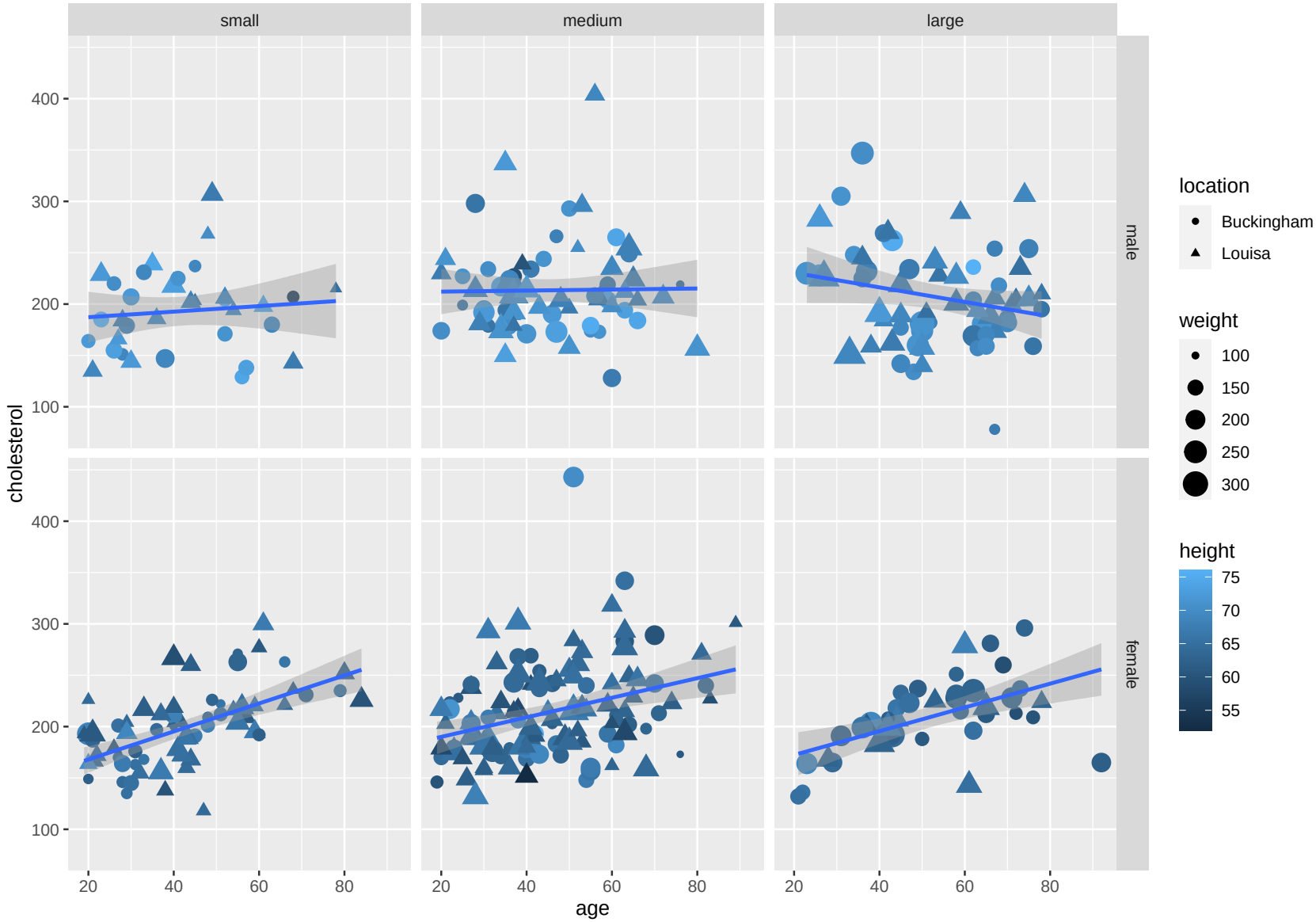
```
a <- ggp + facet_wrap(~Month,scales="free")
b <- ggp + facet_wrap(~Month,nrow=1,scales="free")
grid.arrange(a,b,nrow=2,ncol=1)
```



facet_grid with two grouping variables

- facet_grid(gender frame): separates the plots into a different scatterplot for males/females (as rows) and for each of the three body fits

```
diab <- na.omit(diabetes[,c("chol","weight","gender","frame","age",
                            "height","location")]) # faraway package
ggplot(diab,aes(x=age,y=chol)) +
  geom_point(aes(shape=location,size=weight,col=height)) +
  facet_grid(gender~frame) + geom_smooth(method="lm") + labs(y="cholesterol")
```


small
medium
large
male
female
cholesterol
age
400
300
200
100
20
40
60
80
location
Buckingham
Louisa
weight
100
150
200
250
300
height
75
70
65
60
55

Interactive Tools in ggvis

The ggvis package enables you to design flexible statistical plots that the end user can interact with.

The results are provided as web graphics. When you consider a dynamic experience for visual data exploration, this package will be useful.

```
# simple example of "ggvis"
surv <- na.omit(survey[,c("Sex","Wr.Hnd","Height","Smoke","Exer")])
surv %>% ggvis(x=~Height) %>%
  layer_histograms(width=input_slider(1,15,label="Binwidth:"),fill:="gray")
```

Note: Currently, the Shiny package is more popular than the ggvis. For more details, refer to https://ggvis.rstudio.com/ and https://shiny.rstudio.com/.

Reference

- Davies, T. M. The Book of R. No Starch Press. Chapter 24.