How do we analyze data?

Whether you are an expert in statistics or simply utilizing it for your science, graphing needs to take place at the:

initiation of your data analysis

throughout the data analysis process

presentation of the results from your data analysis
Chicago Bears Private Seat Licenses (PSLs)

The critical features of Trellis Graphics within the lattice package in R will be explored with the Chicago Bears Private Seat Licenses (PSLs) data set.

These PSLs are bought and sold in a market place run by the Bears organization similar to that of pork bellies on the Chicago Mercantile Market Place.
Soldier Stadium Diagram Explained

Each block is a section with its number designation inside.

Groups of sections form areas that are designated by possessing the same filled in or outlined coloring scheme.
<table>
<thead>
<tr>
<th>Color</th>
<th>Sections</th>
<th>2013 Price</th>
<th>2013 Single Game Price*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Club 1</td>
<td>$395</td>
<td>$420</td>
</tr>
<tr>
<td></td>
<td>Club 2</td>
<td>$345</td>
<td>$370</td>
</tr>
<tr>
<td></td>
<td>Club 3</td>
<td>$265</td>
<td>$290</td>
</tr>
<tr>
<td></td>
<td>100-Level Mezzanine; 200-Level Media Deck</td>
<td>$165</td>
<td>$190</td>
</tr>
<tr>
<td></td>
<td>100-Level Mezzanine; 200-Level Media Deck</td>
<td>$145</td>
<td>$170</td>
</tr>
<tr>
<td></td>
<td>100-Level Mezzanine</td>
<td>$138</td>
<td>$163</td>
</tr>
<tr>
<td></td>
<td>300-Level Grandstand; 200-Level Media Deck</td>
<td>$135</td>
<td>$160</td>
</tr>
<tr>
<td></td>
<td>300-Level Grandstand</td>
<td>$125</td>
<td>$150</td>
</tr>
<tr>
<td></td>
<td>300-Level Grandstand</td>
<td>$115</td>
<td>$140</td>
</tr>
<tr>
<td></td>
<td>100-Level Corner Mezzanine</td>
<td>$118</td>
<td>$143</td>
</tr>
<tr>
<td></td>
<td>100-Level Endzone</td>
<td>$103</td>
<td>$128</td>
</tr>
<tr>
<td></td>
<td>400-Level Grandstand (Through Row 15)</td>
<td>$99</td>
<td>$124</td>
</tr>
<tr>
<td></td>
<td>400-Level Grandstand (Above Row 15)</td>
<td>$95</td>
<td>$120</td>
</tr>
<tr>
<td></td>
<td>200-Level Endzone</td>
<td>$93</td>
<td>$118</td>
</tr>
<tr>
<td></td>
<td>300-Level Endzone</td>
<td>$83</td>
<td>$108</td>
</tr>
<tr>
<td></td>
<td>400-Level Grandstand (Through Row 15)</td>
<td>$83</td>
<td>$108</td>
</tr>
<tr>
<td></td>
<td>400-Level Grandstand (Above Row 15)</td>
<td>$79</td>
<td>$104</td>
</tr>
</tbody>
</table>
Where within Soldier Stadium PSLs are Sold

PSLs are intended to be premium seats.

Thus, only seats within certain areas within Soldier Stadium are allowed to be put on the Chicago Bears PSL marketplace.
PSL Data Set

> names(rawpsl.df)
[1] "day" "month" "year" "num" "area" "sec"
[7] "row" "total" "perseat" "price" "log2perseat"

> dim(rawpsl.df)
[1] 988 11

> head(rawpsl.df)
      day month year num area   sec   row total  perseat  price log2perseat
1       1    28  2007   4 265sc3 304    10   20000     5000   265 12.287712
2       2     6   9  2007   2 335mc3 307    16    9900   4950 335 12.273213
3       3    12   9  2007   2 265sc2 203    14 22000 11000 265 13.425224
4       4    12   9  2007   2 265sc2 204     6 11000  5500 265 12.425225
5       5    12   9  2007   2 335mc2 212     3 33000 16500 335 14.010188

> levels(area)
[1] "90e2" "108a1" "110s3" "120m3" "125p3a2" "128s1" "135m1" "135ms2" "265sc2" "265sc3" "335mc2" "335mc3" "385pc23"
Section 4.1

General Display Functions (GDFs):

xyplot()  histogram()
bwplot()  densityplot()
stripplot()  splom()
qq()  parallel()
dotplot()  contourplot()
barchart()  levelplot()
piechart()  wireframe()
qqmath()  cloud()
Sections 4.3; 4.6

GDFs can accept Panel Functions within their arguments including Data Structures that can incorporate critical graphing elements within one function wrapper including, but not limited to:

```
lm() via panel.lmline(x, y, ...)
loess() via panel.loess(x, y, ...)
```

```R
glm()
aov()
```
# Section 5.2
xyplot(rawpsl.df$log2perseat ~ rawpsl.df$row)

# Section 5.3
attach(rawpsl.df)
xyplot(log2perseat ~ row)

# OR
xyplot(log2perseat ~ row, data = rawpsl.df)
data= is the first input to be searched in the path

subset= requires the data= call and will restrict the data to a prescribed subset
Section 5.4

```r
xyplot(formula = log2perseat ~ row,
       data = rawpsl.df,
       subset = price > 135)
```
Section 6.2

```r
xyplot(formula = log2perseat ~ row,
        data = rawpsl.df,
        aspect = "xy")
```
xyplot(formula = log2perseat ~ row,
data = rawpsl.df,
aspect = 1)
bwplot(formula = area ~ log2perseat, 
data = rawpsl.df, 
aspect = 1)
stripplot(formula = area ~ log2perseat,
data = rawpsl.df,
aspect = 1)
qq(formula = area ~ log2perseat, 
data = rawpsl.df, 
subset = (area == "135m1") | (area == "135ms2"), 
aspect = 1)
Section 7.6

dotplot(tapply(log2perseat, area, median),
        aspect = 1,
        cex = 1.25)
barchart(tapply(log2perseat, area, median),
    aspect = 1)
pie(tapply(log2perseat, area, median),
    aspect = 0.5)
Section 7.9

\begin{verbatim}
qqmath(formula = ~perseat,
data = rawpsl.df,
aspect = 1)
\end{verbatim}
histogram(formula = ~perseat,
data = rawpsl.df,
aspect = 1,
nint = 10)
densityplot(formula = ~perseat, 
data = rawpsl.df, 
aspect = 1)
splom(formula = ~rawpsl.df[c(3:5, 7, 11)])
parallel(formula = ~rawpsl.df[c(3:5, 7, 11)])
Section 7.15

```r
contourplot(formula = log2perseat ~ year*row,
data = rawpsl.df,
aspect = 1,
at = seq(10, 15, by = 0.5))
```
levelplot(formula = log2perseat ~ year*row, 
data = rawpsl.df, 
aspect = 1, 
cuts = 6)
wireframe(formula = log2perseat ~ year*row, 
data = rawpsl.df)
cloud(formula = log2perseat ~ year*row, 
data = rawpsl.df)
Section 8.1

\[
\begin{align*}
\text{perseat.qqmath} & \leftarrow \text{qqmath}(\text{formula} = \sim \text{perseat}, \\
& \quad \text{data} = \text{rawpsl.df}) \\
\text{log2perseat.qqmath} & \leftarrow \text{qqmath}(\text{formula} = \sim \log2\text{perseat}, \\
& \quad \text{data} = \text{psl.rawsold.df}) \\
\text{print(} \text{perseat.qqmath, position} = \text{c(0,0,1,0.4)}, \text{ more=} T\text{)} \\
\text{print(} \text{log2perseat.qqmath, position} = \text{c(0,0.35,1,1)}\text{)}
\end{align*}
\]
The operator to queue Multipanel Conditioning is the familiar probabilist operator to indicate conditional or "given" information of "|".
xyplot(formula = log2perseat ~ row | area,
       data = psl.rawsold.df)
log2byareaqq <- qqmath(formula=\~\log2\text{per seat} | \text{area},
data = psl.rawsold.df,
layout = c(4, 4))

log2byareaqq
Section 9.11

```r
xyplot(formula = log2perseat ~ row | year,
       data = psl.rawsold.df,
       aspect = 0.5)
```
GIVEN.row <- equal.count(row, number = 9, overlap = 0)

xyplot(formula = log2perseat ~ year | GIVEN.row,
       data = psl.rawsold.df,
       aspect = 2.5)
GIVEN.row <- equal.count(row, number = 9, overlap=0)
plot(GIVEN.row)
dotplot(tapply(log2perseat, area, median),
  main = "dotplot of log2perseat medians by area",
  ylab = "area",
  xlab = "log2perseat (units = log2$)"")
dotplot of log2perseat medians by area

log2perseat (units = log2$) vs. area

90e2
108a1
110s3
120m3
125p3a2
128s1
135m1
135ms2
265sc2
265sc3
335mc2
335mc3
385pc23
11121314
Section 11.1 Three Kick Methods

Three Kick Methods:

send another graph

dev.off()

q()
Section 12.3

xyplot(formula = log2perseat ~ row,
data = psl.rawsold.df,
main = "Scatter plot of log2perseat vs. row",
ylab = "log2perseat (units = log2$)",
panel = function(x, y)
{
  biggest <- y == max(y)
  panel.points(x[!biggest], y[!biggest], pch = ".")
  panel.points(x[biggest], y[biggest], pch = "M")
})
Scatter plot of log2perseat vs. row

log2perseat (units = log2$)

row
Section 12.4

```r
xyplot(formula = log2perseat ~ row | area,
data = psl.rawsold.df,
main = "Scatter plot of log2perseat vs. row by area",
ylab = "log2perseat (units = log2$)",
panel = function(x, y)
{
  biggest <- y == max(y)
  panel.points(x[!biggest], y[!biggest], pch = ".")
  panel.points(x[biggest], y[biggest], pch = "M")
})
```
Section 12.5

```r
xyplot(formula = log2perseat ~ row | area,
data = psl.rawsold.df,
main = "Scatter plot of log2perseat vs. row by area",
ylab = "log2perseat (units = log2$)",
skip = c(rep(F, 8), rep(T, 2)),
layout = c(5, 3),
panel = function(x, y)
{
  panel.points(x, y, pch = ".", col = 1, cex = 1.5)
  panel.loess(x, y, span = 1, family = "gaussian")
  panel.lmline(x, y)  # Inserts best line per panel
}
```
Chapter 12 Panel Functions

Core graphics functions commonly used in writing panel functions:

points()
lines()
text()
segments()
polygon()

Core parameters commonly used in writing panel functions:

col
lty
pch
lwd
cex

Use ?par for their definitions.
xyplot(formula = log2perseat ~ row, 
data = psl.rawsold.df, 
subset = price > 135, 
main = "Scatter plot of log2perseat vs. row grouped by area club subset",
ylab = "log2perseat (units = log2$)", 
groups = area, 
panel = panel.superpose)
Scatter plot of log2perseat vs. row grouped by area club subset
Section 14.1

clubsymbols <- c("s2", "s3", "m2", "m3", "p")

xyplot(formula = log2perseat ~ row,
data = psl.rawsold.df,
subset = price > 135,
main = "Scatter plot of log2perseat vs. row grouped by area club subset",
ylab = "log2perseat (units = log2$)",
pch = clubsymbols,
groups = area,
panel = panel.superpose)
clubsymbols <- c("2", "3", "2", "3", "p")
clubcol <- c("blue", "blue", "red", "red", "purple")

xyplot(formula = log2perseat ~ row,
data = psl.rawsold.df,
subset = price > 135,
main = "Scatter plot of log2perseat vs. row grouped by area club subset",
ylab = "log2perseat (units = log2$)",
pch = clubsymbols,
col = clubcol,
groups = area,
panel = panel.superpose)
Scatter plot of log2perseat vs. row grouped by area club subset

log2perseat (units = log2$)

row
sections 13.3; 14.2

clubsymbols <- c("2", "3", "2", "3", "p")
clubcol <- c("blue", "blue", "red", "red", "purple")

superpose.symbol <- trellis.par.get("superpose.symbol")
superpose.symbol$pch[1:5] <- clubsymbols
superpose.symbol$col[1:5] <- clubcol
trellis.par.set("superpose.symbol", superpose.symbol)
superpose.symbol <- trellis.par.get("superpose.symbol")

xyplot(formula = log2perseat ~ row,
data = psl.rawsold.df,
subset = price > 135,
main = "Scatter plot of log2perseat vs. row grouped
by area club subset",
ylab = "log2perseat (units = log2$)",
pch = clubsymbols,
col = clubcol,
groups = area,
panel = panel.superpose,
key = list
  (  
    points = Rows(superpose.symbol, 1:5),
    text = list(levels(area)[9:13])
  ))
Scatter plot of log2perseat vs. row grouped by area club subset

- 265sc2
- 265sc3
- 335mc2
- 335mc3
- 385pc23
Section 14.2

cubsymbols <- c("2", "3", "2", "3", "p")
clubcol <- c("blue", "blue", "red", "red", "purple")

superpose.symbol <- trellis.par.get("superpose.symbol")
superpose.symbol$pch[1:5] <- clubsymbols
superpose.symbol$col[1:5] <- clubcol
trellis.par.set("superpose.symbol", superpose.symbol)
superpose.symbol <- trellis.par.get("superpose.symbol")

xyplot(formula = log2perseat ~ row,
data = psl.rawsold.df,
subset = price > 135,
main = "Scatter plot of log2perseat vs. row grouped
by area club subset",
ylab = "log2perseat (units = log2$)",
pch = clubsymbols,
col = clubcol,
groups = area,
panel = panel.superpose,
key = list
(
    points = Rows(superpose.symbol, 1:5),
    text = list(levels(area)[9:13]),
    space = "right"
))
Section 17.2

```r
xyplot(formula = log2perseat ~ row | area,
data = psl.rawsold.df,
main = "Scatter plot of log2perseat vs. row by area",
ylab = "log2perseat (units = log2$)",
skip = c(rep(F, 8), rep(T, 2)),
layout = c(5, 3),
panel = function(x, y)
{
  panel.points(x, y, pch = ".", col = 1, cex = 1.5)
  panel.loess(x, y, span = 1, family = "gaussian")
  panel.lmline(x, y) # Inserts best line per panel
}
```