NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

5.0 Graphics and Plotting

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

@thephdataminer

Module 6 of the Business Intelligence and Analytics Certification of UP NEC and the UP Center for Business Intelligence

Outline for this Training

- Introduction to R and R Studio
- Data Types and Operators
  - Case Study on R Scripting
- Reading, Manipulating and Writing Data
  - Case Study on Dataset Analysis with ETL
- Basic R Programming
  - Case Study: Writing Functions
- **Graphics and Plotting**
- Deploying R and Dashboard Generation
  - Case Study: Deploying a Simple Dashboard
- Deploying R with C#
  - Case Study: A Simple Standalone GUI For R Apps

Outline for this Session

- Plotting Commands
- Specialized Plotting Commands
- Plotting Arguments
- ggplot

Plotting Commands

- Graphical facilities are an important and extremely **versatile component** of the R environment.
- It is possible to use the facilities to display a wide variety of **statistical graphs** and also to build entirely new types of graph

Plotting Commands

Definition 5.1: High-Level Functions

- High-level plotting functions **create a new plot** on the graphics device, possibly with axes, labels, titles and so on.

Definition 5.2: Low-Level Functions

- Low-level plotting functions add more information **to an existing plot**, such as extra points, lines and labels.
- R maintains a list of graphical parameters which can be manipulated to customize your plots.

Plotting Commands

- One of the most frequently used plotting functions in R is the plot() function.
- This is a generic function: the type of plot produced is dependent on the type or class of the first argument

Plotting Commands

Definition 5.3: The plot() function

- `plot(x, y)`
- `plot(xy)`
- If x and y are vectors, `plot(x, y)` produces a scatterplot of y against x.
- The same effect can be produced by supplying one argument as either a list containing two elements x and y or a two-column matrix.

Plotting Commands

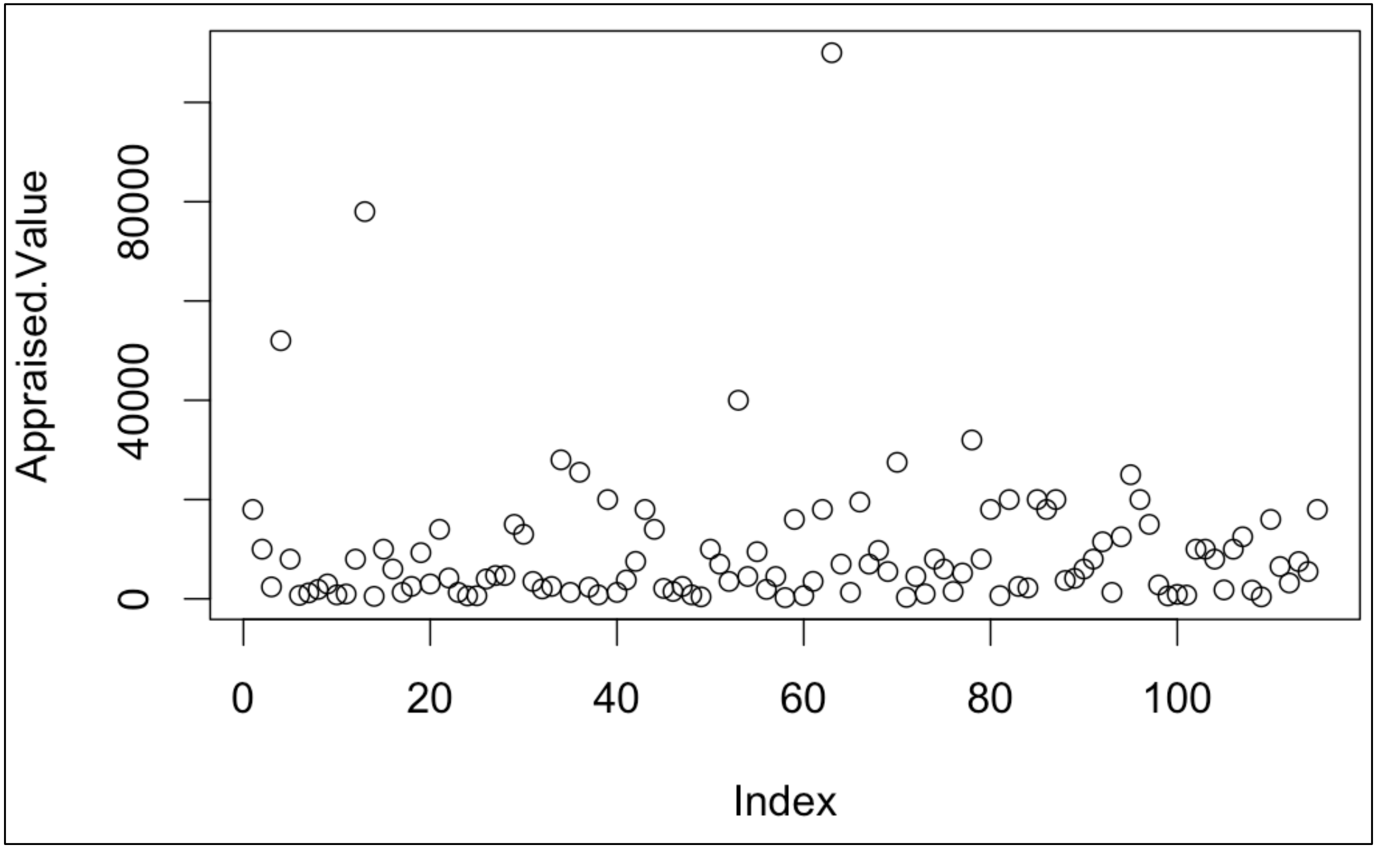
- `plot(x)`
- If x is a time series, this produces a time-series plot. If x is a numeric vector, it produces a plot of the values in the vector against their index in the vector
- `plot(f)`
- f is a factor object, y is a numeric vector. This generates a bar plot of f
- `plot(f, y)`
- produces boxplots of y for each level of f.

Plotting Commands

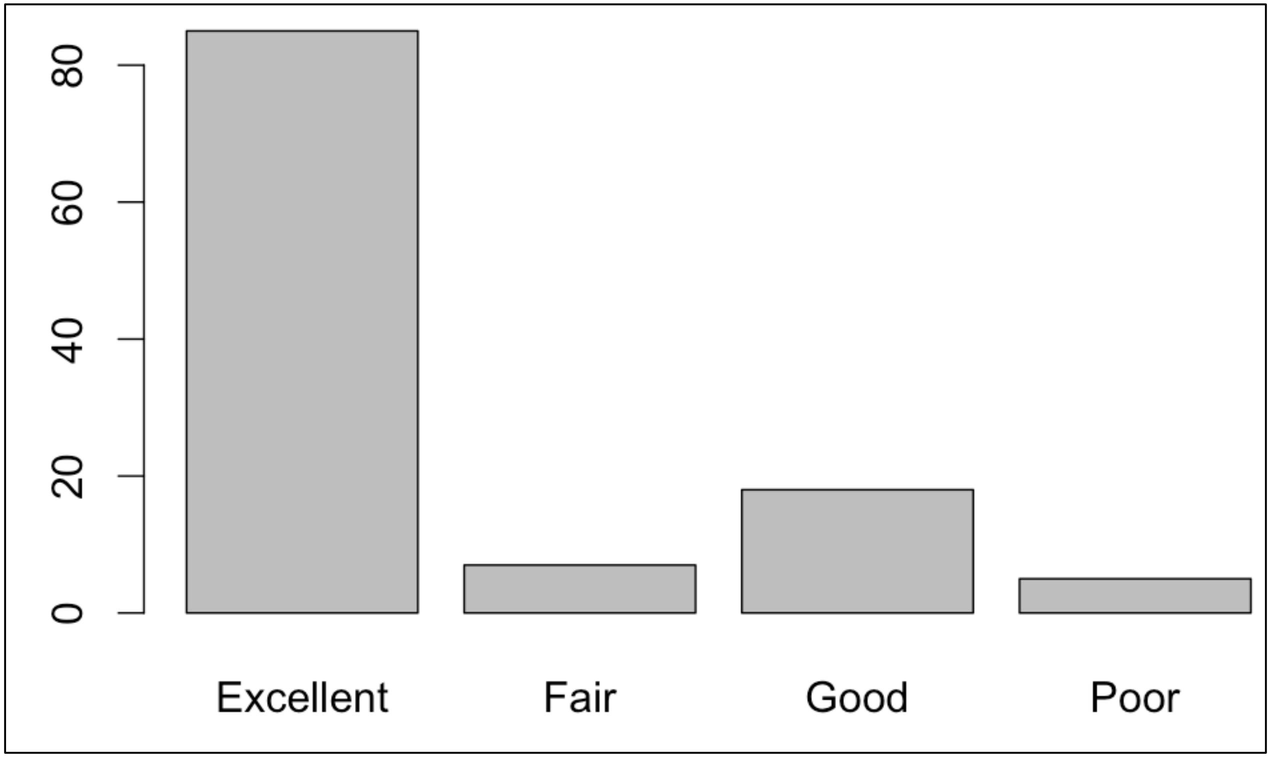
Example 5.1: Plotting

```
options("scipen"=100, "digits"=4)
plotdata<-
read.csv(file=”nSQLExtract.csv",head=TRUE,sep=",")
attach(plotdata)
#Plot Appraised Value Only
plot(Appraised_Value)
#Bar Graph of Condition
plot(Condition)
#Box Plot of Condition by Appraised Value
plot(Condition,Appraised_Value)
```

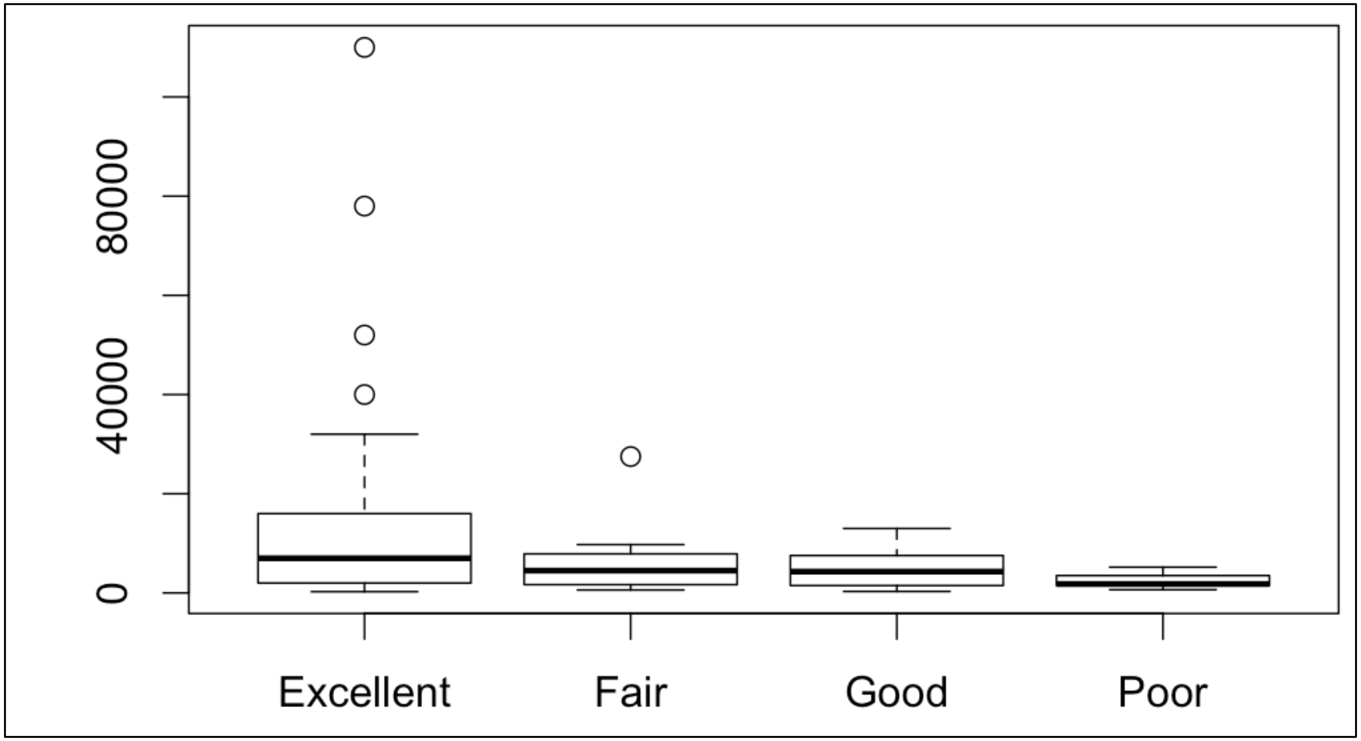
Plotting Commands



Plotting Commands



Plotting Commands



Outline for this Session

- Plotting Commands
- **Specialized Plotting Commands**
- Plotting Arguments
- ggplot

Specialized Plotting Commands

- Displaying Multivariate Data
  - If X is a numeric matrix or data frame, the command
  - `pairs(X)`
  - produces a pairwise scatterplot matrix of the variables defined by the columns of X, that is, every column of X is plotted against every other column of X and the resulting n(n - 1) plots are arranged in a matrix with plot scales constant over the rows and columns of the matrix.

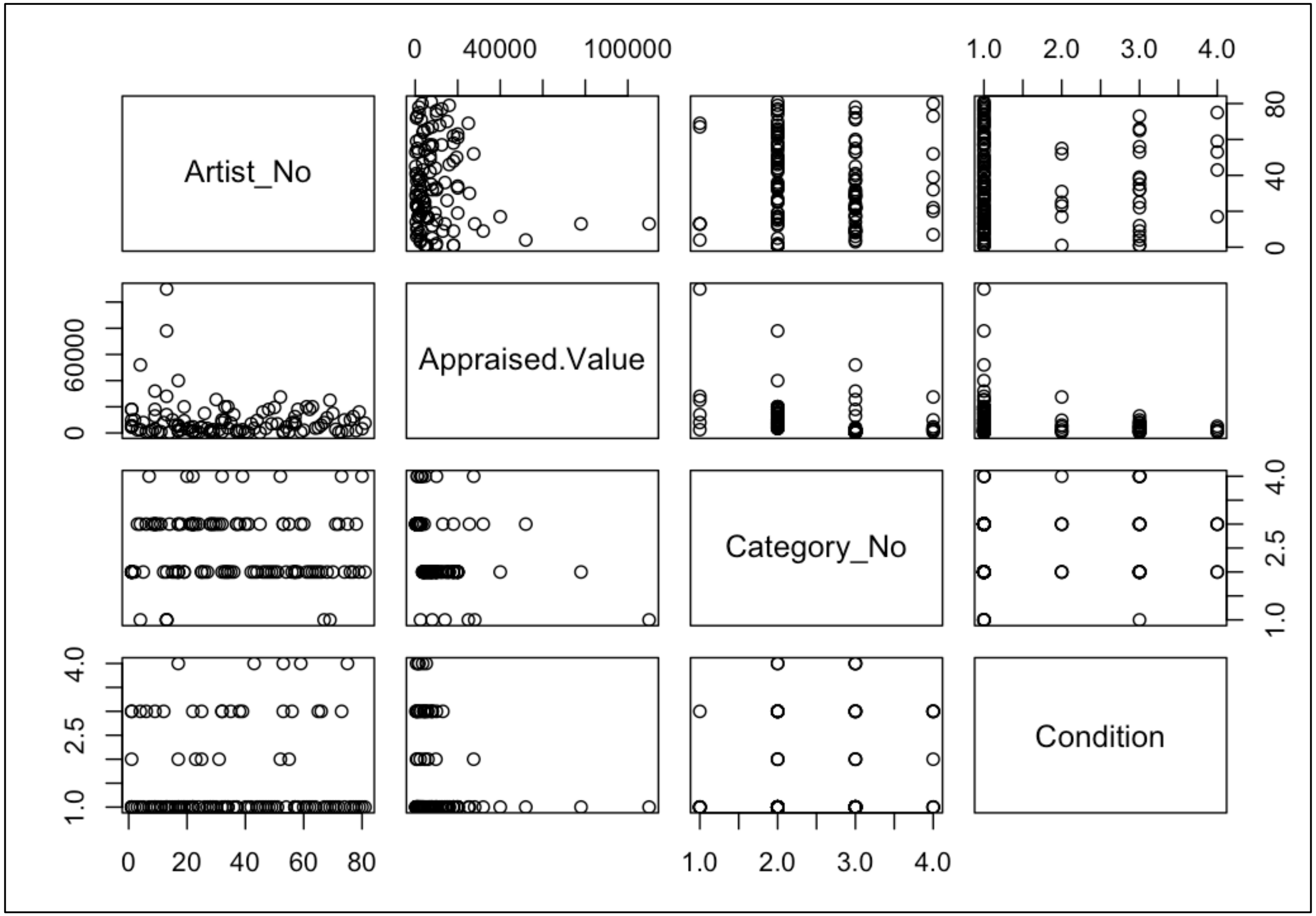
Specialized Plotting Commands

Example 5.2: Plotting a Scatterplot

```
#Plotting a Scatterplot
```

```
pairs(plotdata[,c("Artist_No","Appraised_Value","Category_No","Condition")])
```

Specialized Plotting Commands



Specialized Plotting Commands

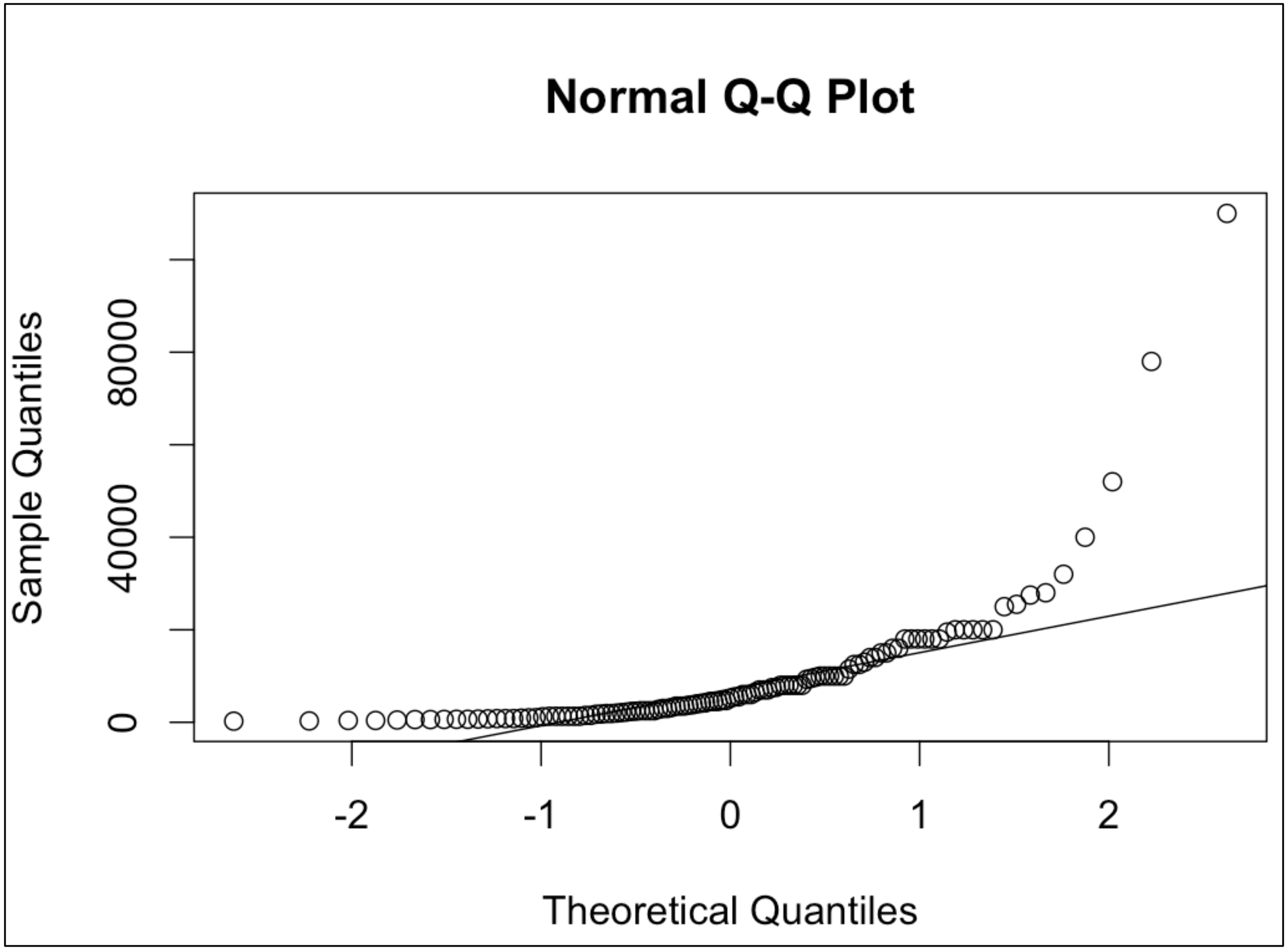
- Distribution-Comparison Plots
  - `qqnorm(x)`
  - Plots the numeric vector x against the expected Normal order scores (a normal scores plot)
  - `qqline(x)`
  - Adds a straight line to such a plot by drawing a line through the distribution and data quartiles.
  - `qqplot(x, y)`
  - Plots the quantiles of x against those of y to compare their respective distributions.

Specialized Plotting Commands

Example 5.3: Distribution Comparison Plots

- `qqline(Appraised_Value)`

Specialized Plotting Commands



Specialized Plotting Commands

➢ `hist(x)`

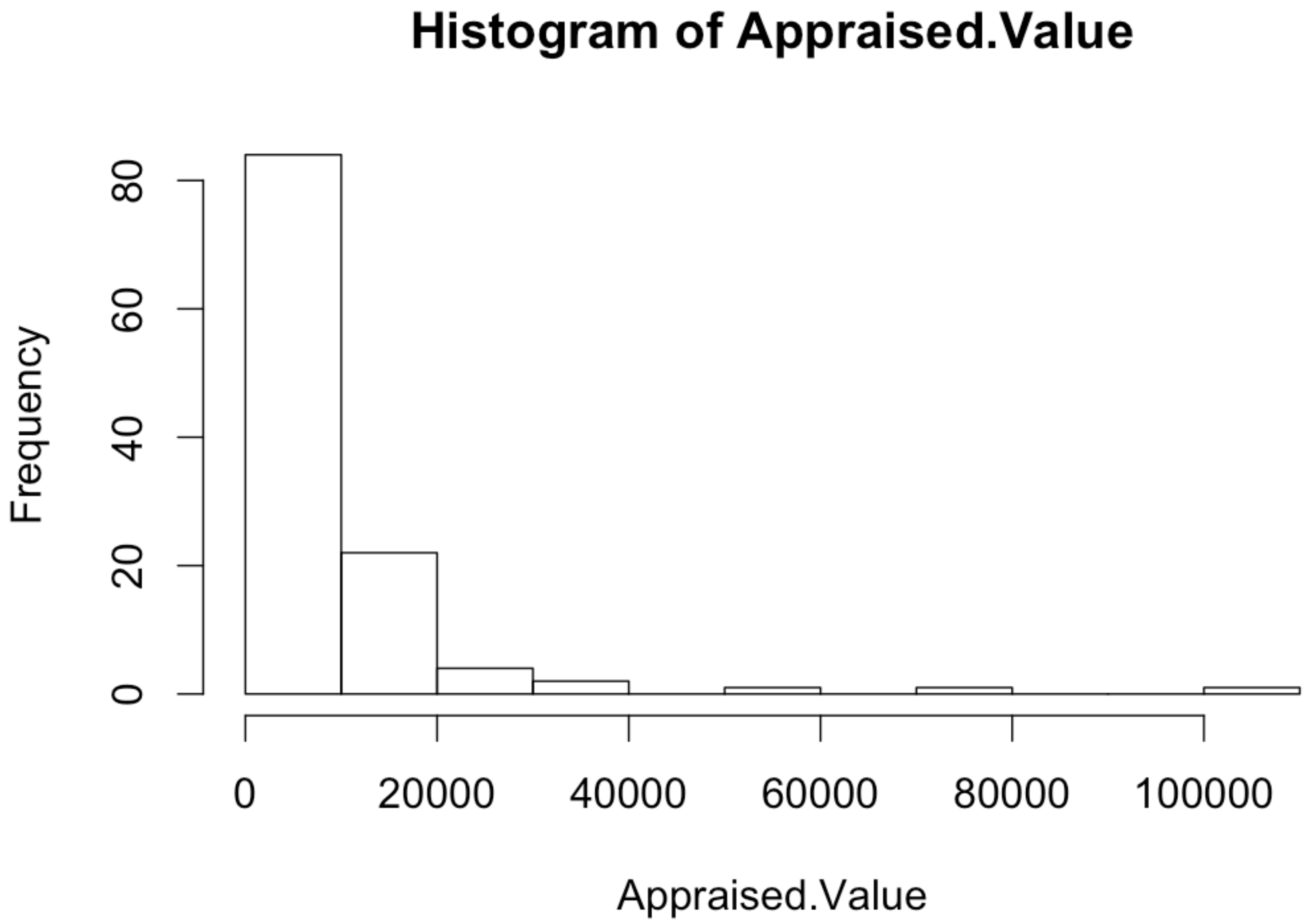
- Produces a **histogram** of the numeric vector x.
- `hist(x, nclass=n)`
- A sensible **number of classes** is usually chosen, but a recommendation can be given with the nclass= argument.

Specialized Plotting Commands

Example 5.4: Histogram

- `hist(Appraised_Value)`

Specialized Plotting Commands



Specialized Plotting Commands

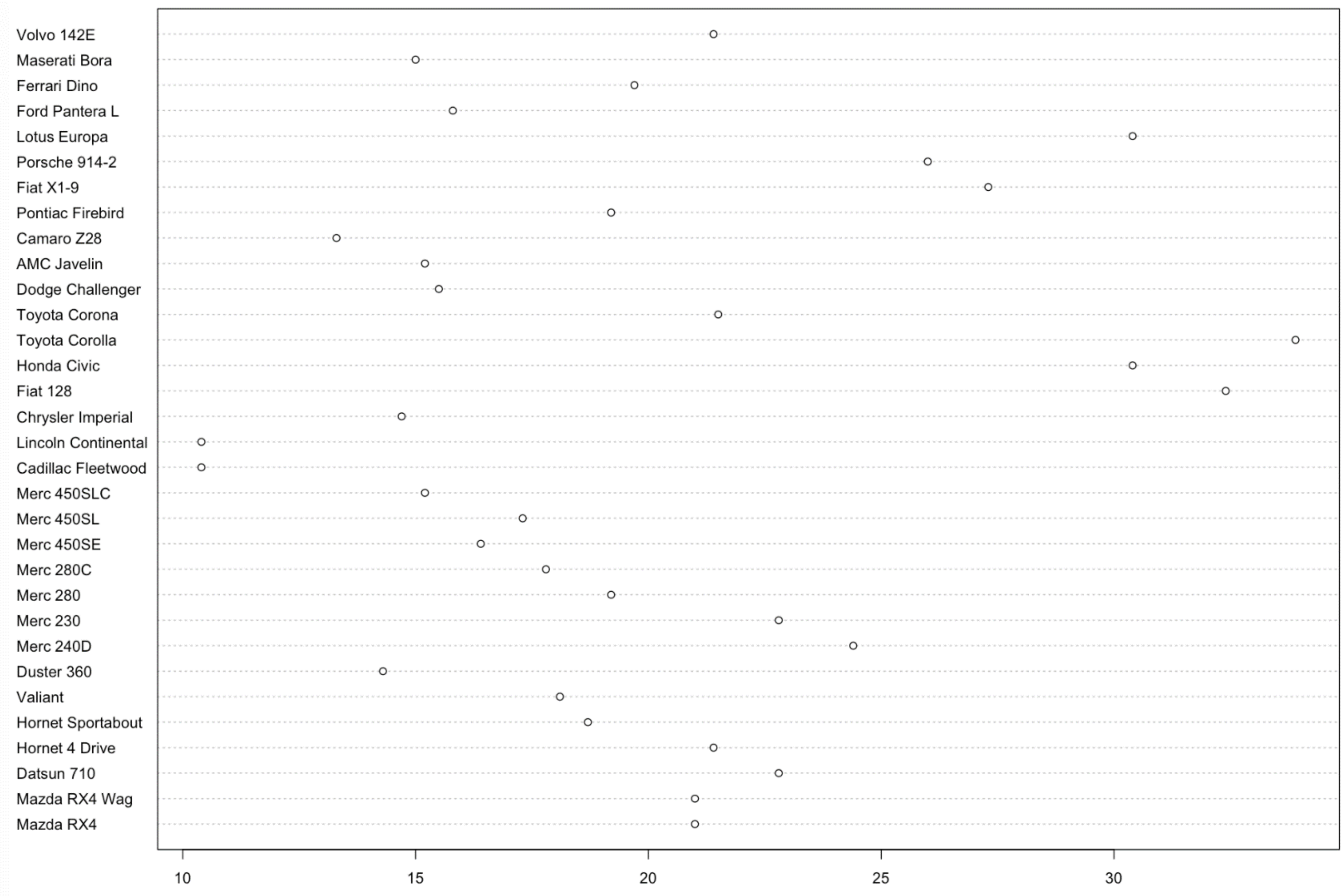
- `dotchart(x, ...)`
- Constructs a **dotchart** of the data in x. In a dotchart the y-axis gives a labelling of the data in x and the x-axis gives its value.
- For example it allows easy visual selection of all data entries with values lying in specified ranges.

Specialized Plotting Commands

Example 5.5: Dot Charts

- `dotchart(mtcars$mpg,labels=row.names(mtcars),cex=.7)`

Specialized Plotting Commands



Outline for this Session

- Plotting Commands
- Specialized Plotting Commands
- **Plotting Arguments**
- ggplot

Plotting Arguments

- There are a number of arguments which may be passed to high-level graphics functions

➢ `type=`

- – The type= argument controls the type of plot produced, as follows:
    - type="p" Plot individual points (the default)
    - type="l" Plot lines
    - type="b" Plot points connected by lines (both)
    - type="o" Plot points overlaid by lines
    - type="h" Plot vertical lines from points to the zero axis (high-density)

Plotting Arguments

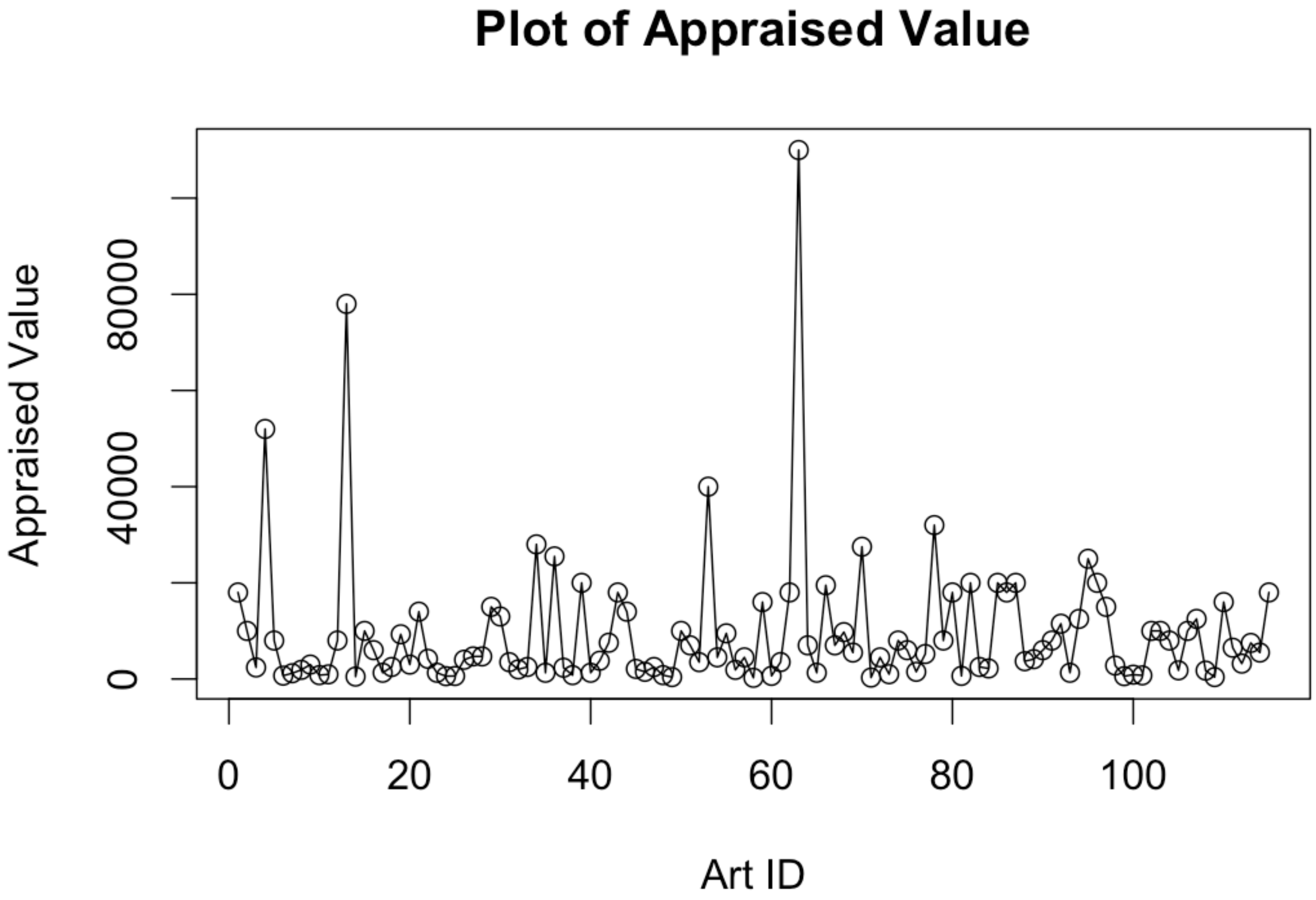
- `main=string`
  - **Figure title,** placed at the top of the plot in a large font.
- `xlab=string`
- `ylab=string`
  - **Axis labels** for the x and y axes. Use these arguments to change the default labels, usually the names of the objects used in the call to the high-level plotting function.

Plotting Arguments

Example 5.6: Plotting Arguments

```
plot(Appraised_Value, type=‘o’,
main='Plot of Appraised Value',
xlab='Art ID', ylab='Appraised Value')
```

Plotting Arguments



Plotting Arguments

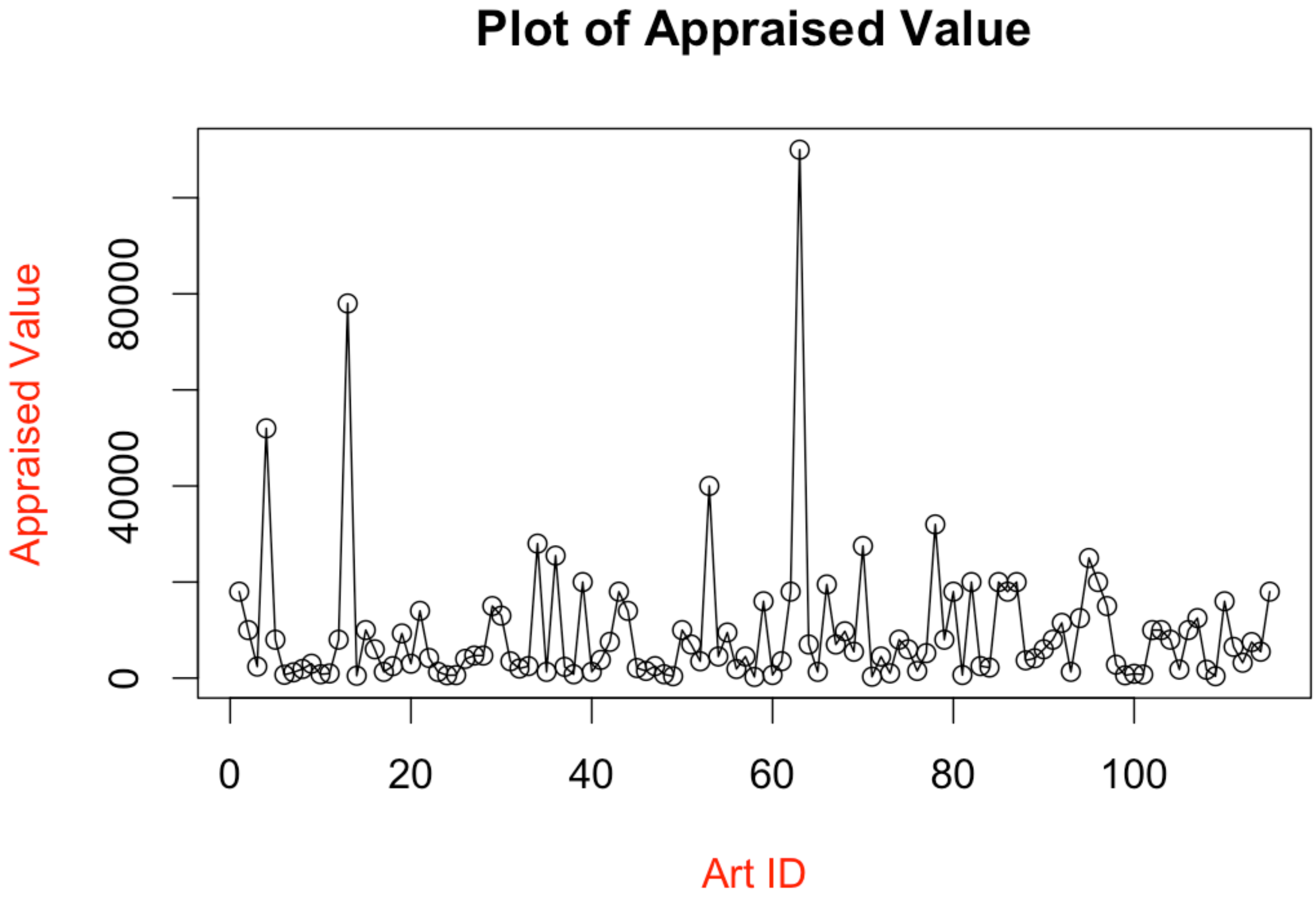
- Set a graphical parameter using par()
- `par()`
  - view current settings
- `opar <- par()`
  - make a copy of current settings
- `par(col.lab="red")`
  - red x and y labels
- `par(opar)`
  - restore original settings

Plotting Arguments

Example 5.7: The **`par`** ( ) function

- `par()`
- `opar <- par()`
- `par(col.lab="red")`
- `plot(Appraised_Value, type=‘o’, main='Plot of Appraised Value', xlab='Art ID', ylab='Appraised Value')`
- `par(opar)`

Plotting Arguments



Plotting Arguments

Option	Description
cex	number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, 0.5 is 50% smaller, etc.
cex.axis	magnification of axis annotation relative to cex
cex.lab	magnification of x and y labels relative to cex
cex.main	magnification of titles relative to cex
cex.sub	magnification of subtitles relative to cex

Plotting Arguments

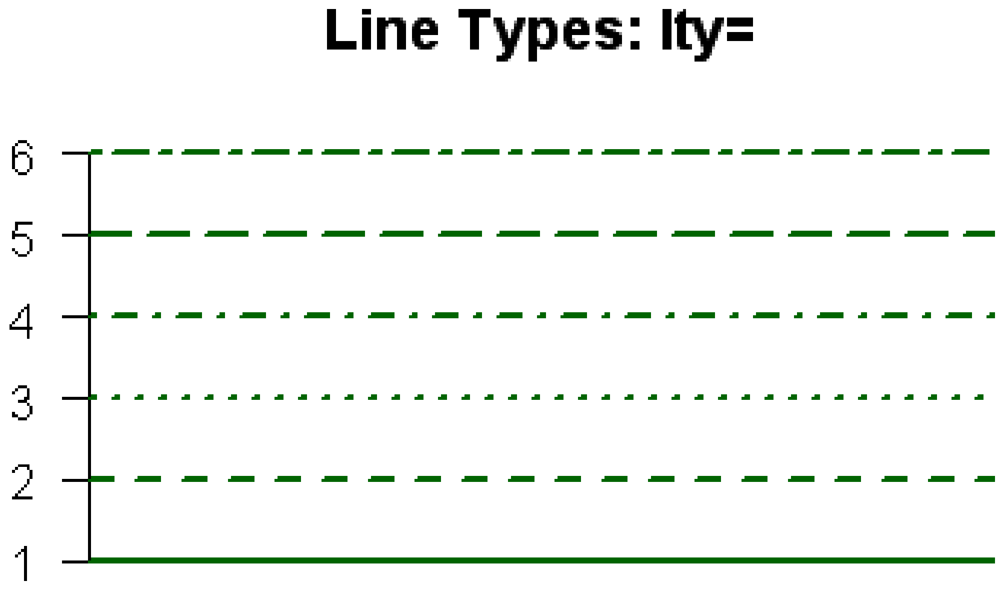
Example 5.8: Text Size

-
  ```
  plot(Appraised_Value, type='o',
  main='Plot of Appraised Value',
  xlab='Art ID', ylab='Appraised Value',
  cex=2.0,cex.main=0.75)
  ```

Plotting Arguments

- Line Type Commands:

Option	Description
lty	line type. see the chart below.
lwd	line width relative to the default (default=1). 2 is twice as wide.

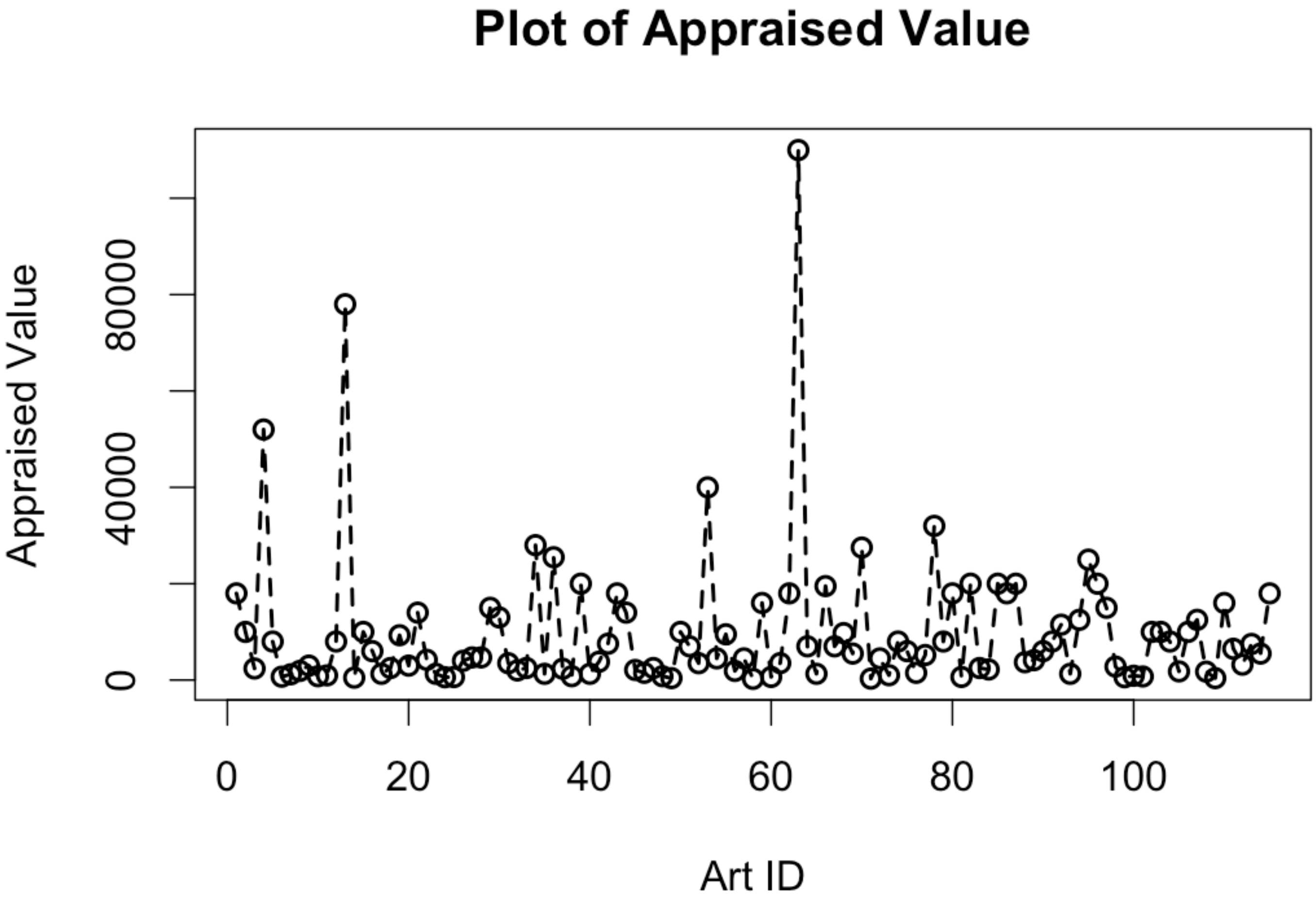


Plotting Arguments

Example 5.9: Line Types

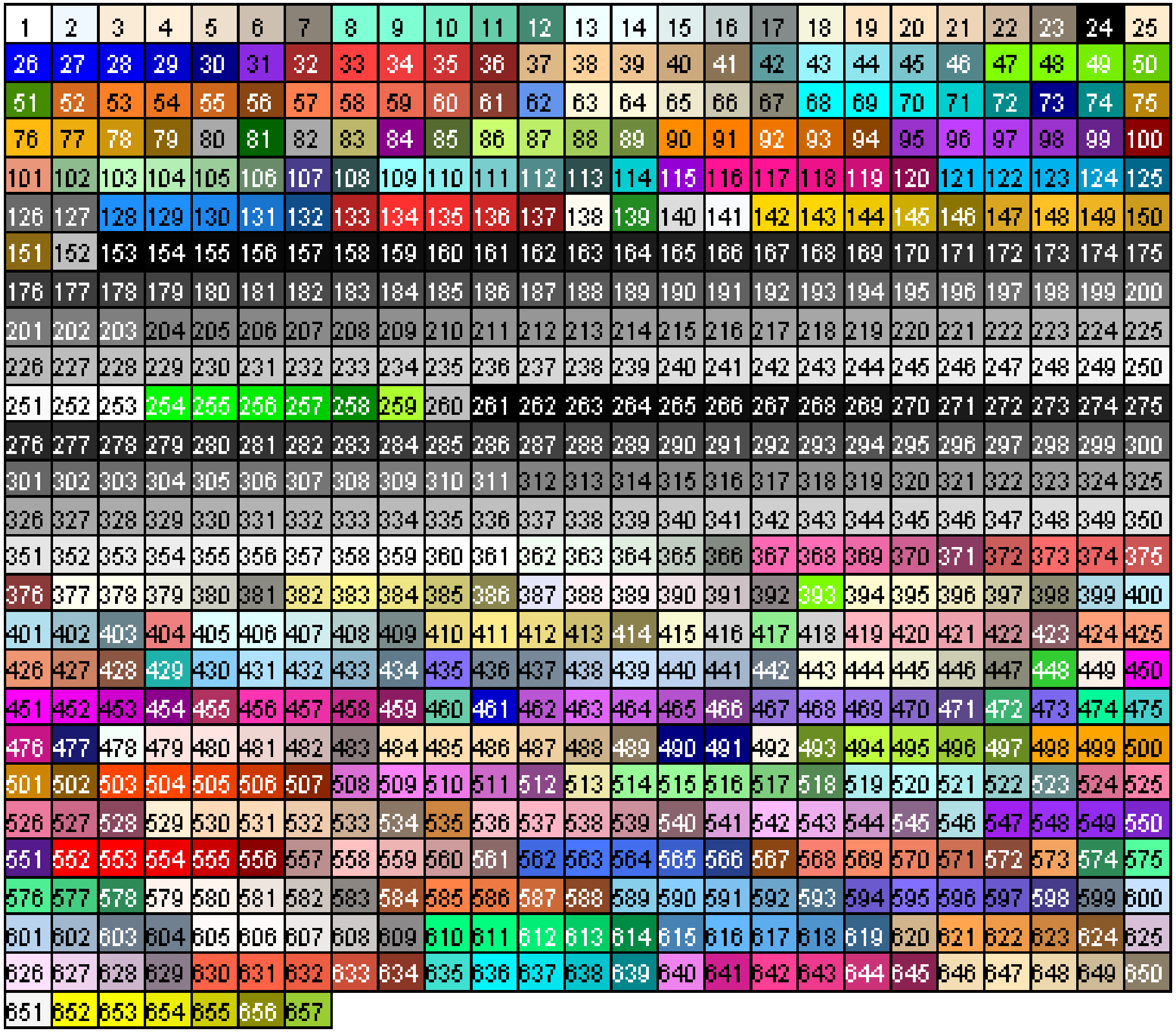
```
plot(Appraised_Value, type='o',
main='Plot of Appraised Value',
xlab='Art ID', ylab='Appraised Value',
lty=2, lwd=2)
```

Plotting Arguments



Plotting Arguments

- You can specify **colors** in **R** by index, name, hexadecimal, or RGB. For example **col= 1**, **col="white ",** and **col="#FF FFFF"** are equivalent



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625
626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650
651	652	653	654	655	656	657																		

Plotting Arguments

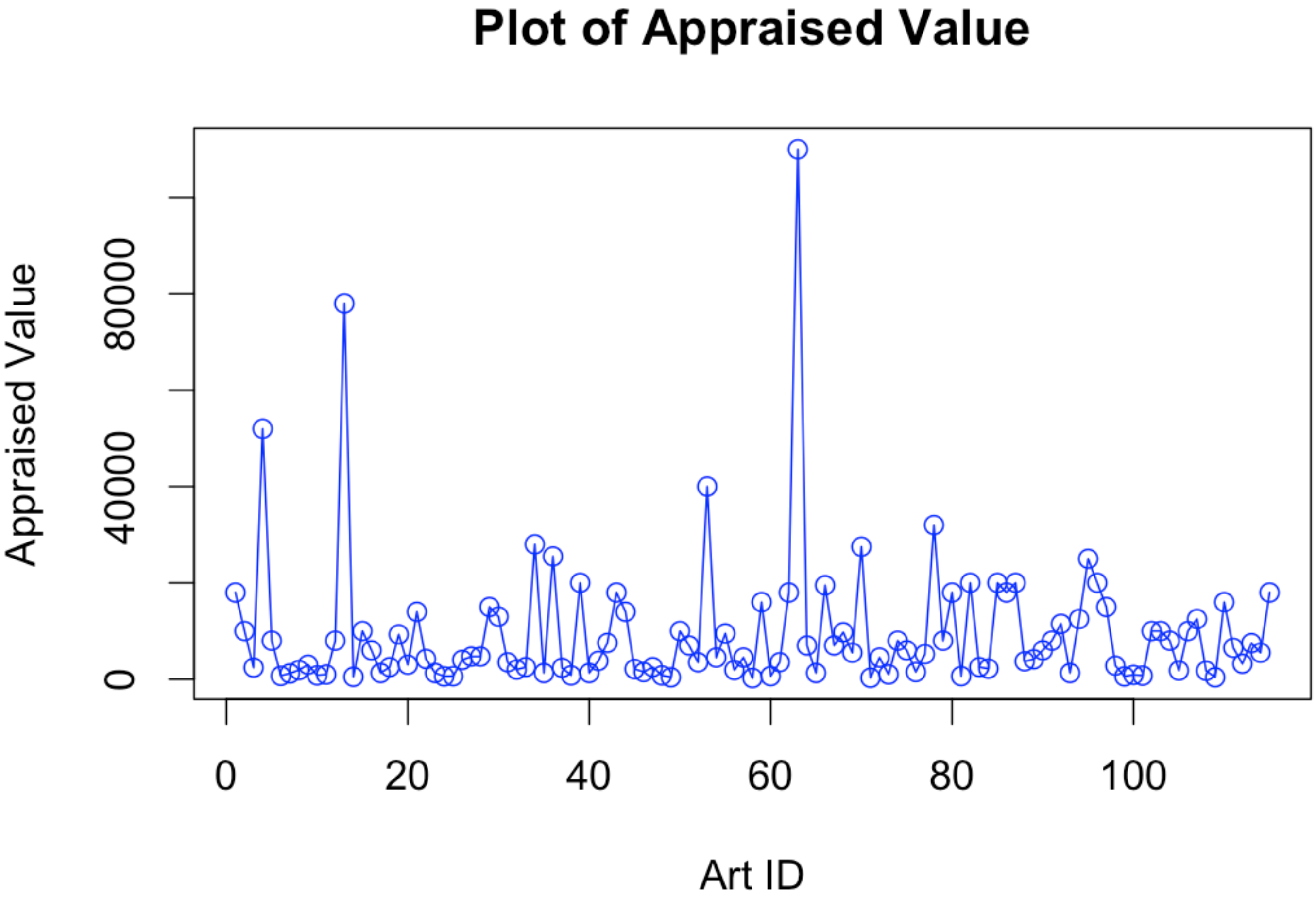
Option	Description
col	Default plotting color. Some functions (e.g. lines) accept a vector of values that are recycled.
col.axis	color for axis annotation
col.lab	color for x and y labels
col.main	color for titles
col.sub	color for subtitles
fg	plot foreground color (axes, boxes - also sets col= to same)
bg	plot background color

Plotting Arguments

Example 5.10: Colors

-
  ```
  plot(Appraised_Value, type='o',
  main='Plot of Appraised Value',
  xlab='Art ID', ylab='Appraised Value',
  col=28)
  ```

Plotting Arguments



Plotting Arguments

- Fonts

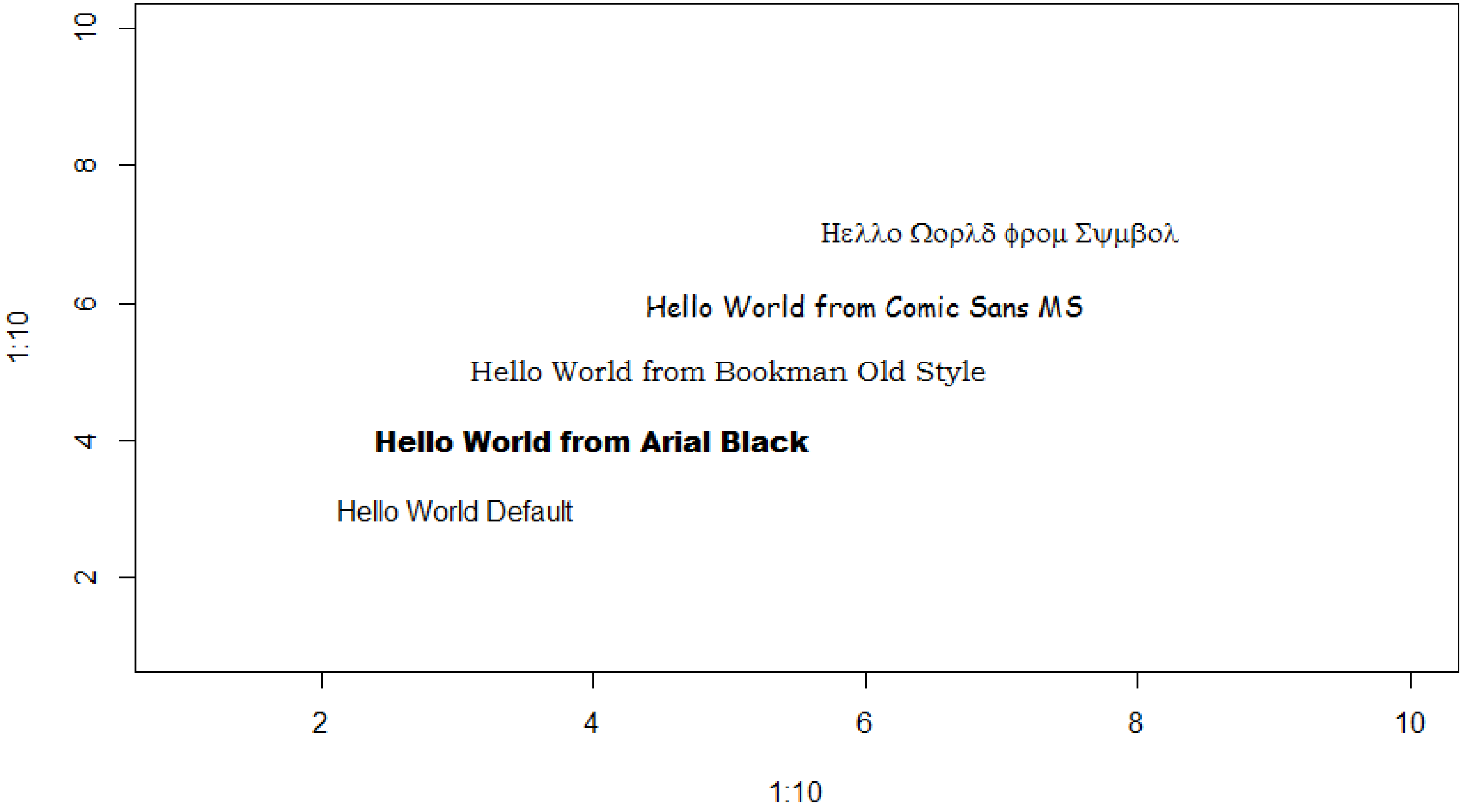
Option	Description
font	Integer specifying font to use for text. 1=plain, 2=bold, 3=italic, 4=bold italic, 5=symbol
font.axis	font for axis annotation
font.lab	font for x and y labels
font.main	font for titles
font.sub	font for subtitles
ps	font point size (roughly 1/72 inch) text size=ps*cex
family	font family for drawing text. Standard values are "serif", "sans", "mono", "symbol". Mapping is device dependent.

Plotting Arguments

Example 5.11: Fonts

```
plot(1:10,1:10,type="n")
windowsFonts(
  A=windowsFont("Arial Black"),
  B=windowsFont("Bookman Old Style"),
  C=windowsFont("Comic Sans MS"),
  D=windowsFont("Symbol")
)
text(3,3,"Hello World Default")
text(4,4,family="A","Hello World from Arial Black")
text(5,5,family="B","Hello World from Bookman Old Style")
text(6,6,family="C","Hello World from Comic Sans MS")
text(7,7,family="D", "Hello World from Symbol")
```

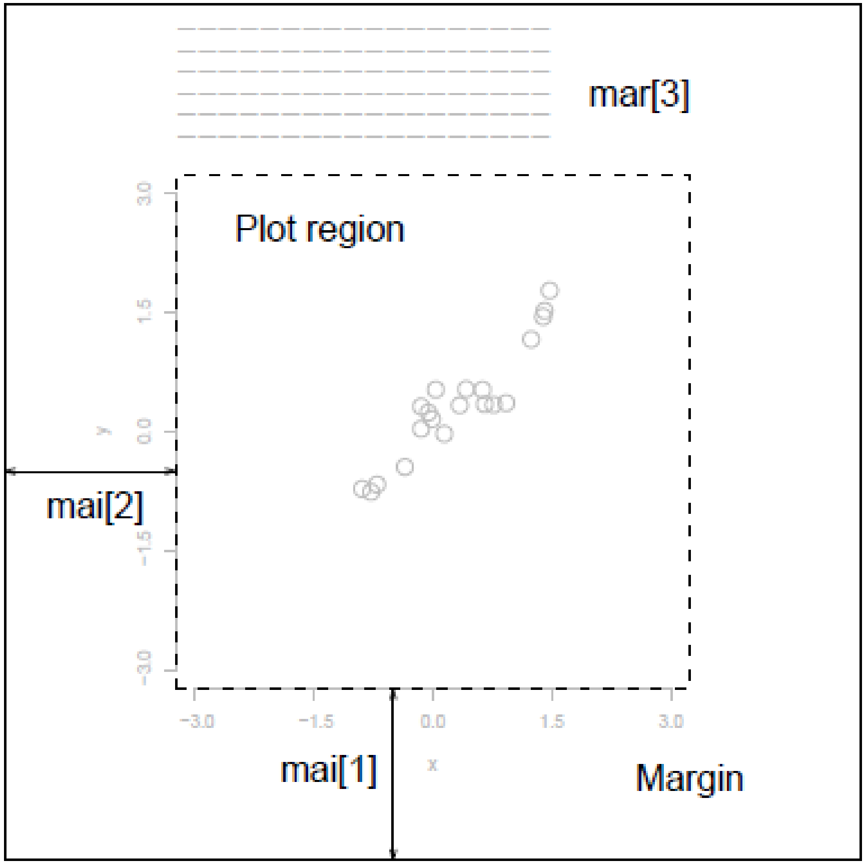
Plotting Arguments



Plotting Arguments

- Margins

Option	Description
mar	numerical vector indicating margin size c(bottom, left, top, right) in lines. default = c(5, 4, 4, 2) + 0.1
mai	numerical vector indicating margin size c(bottom, left, top, right) in inches
pin	plot dimensions (width, height) in inches



Plotting Arguments

- Legends

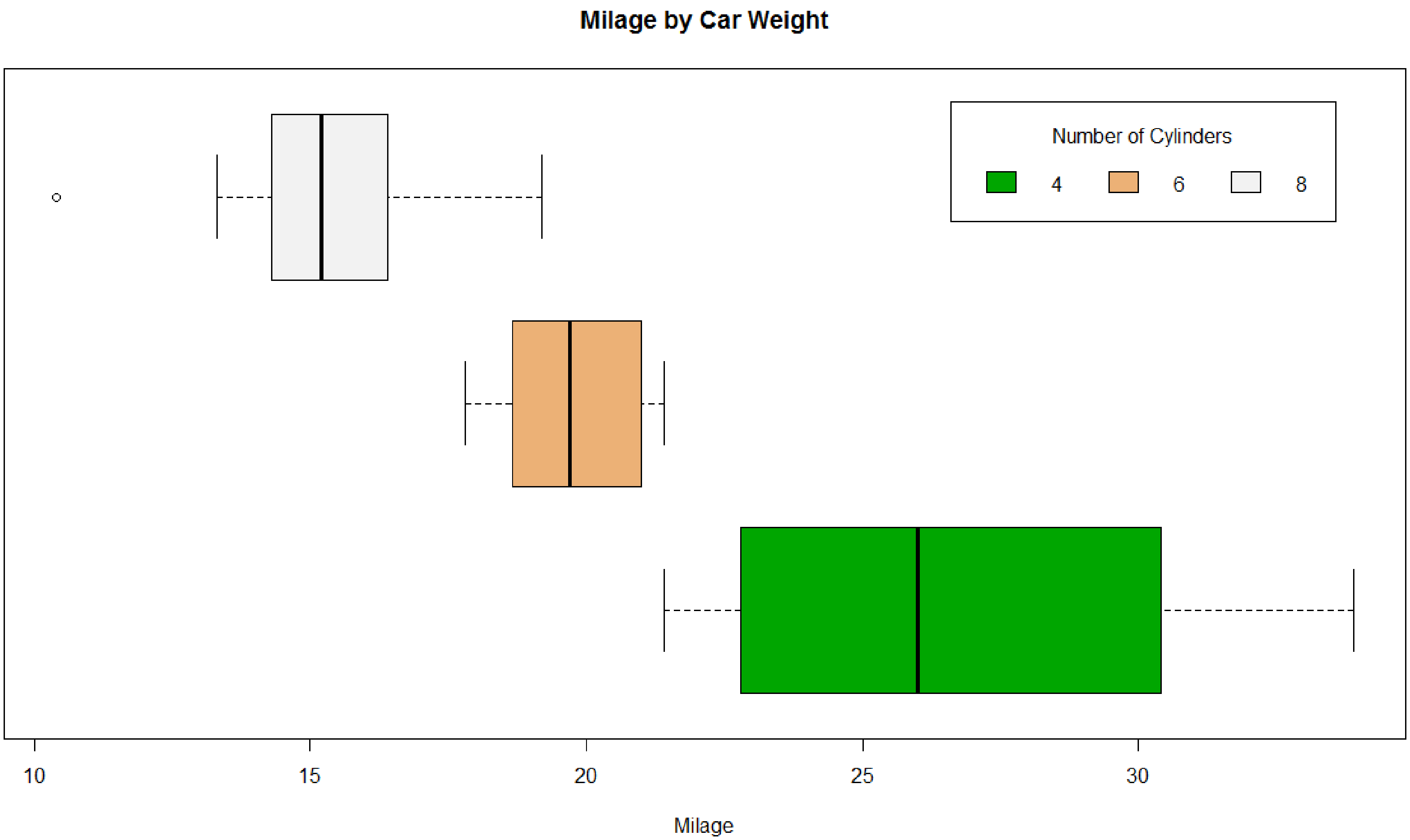
option	**description**
location	There are several ways to indicate the location of the legend. You can give an **x,y coordinate** for the upper left hand corner of the legend. You can use**locator(1)**, in which case you use the mouse to indicate the location of the legend. You can also use the **keywords** "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "bottomright", or "center". If you use a keyword, you may want to use **inset=** to specify an amount to move the legend into the graph (as fraction of plot region).
title	A character string for the legend title (optional)
legend	A character vector with the labels

Plotting Arguments

Example 5.12: Margins and Legends

```
➢ attach(mtcars)
➢ boxplot(mpg~cyl, main="Milage by Car
  Weight", yaxt="n", xlab="Milage",
  horizontal=TRUE,
  col=terrain.colors(3))
➢ legend("topright", inset=.05,
  title="Number of Cylinders",
  c("4","6","8"), fill=terrain.colors(3),
  horiz=TRUE)
```

Plotting Arguments



Plotting Arguments

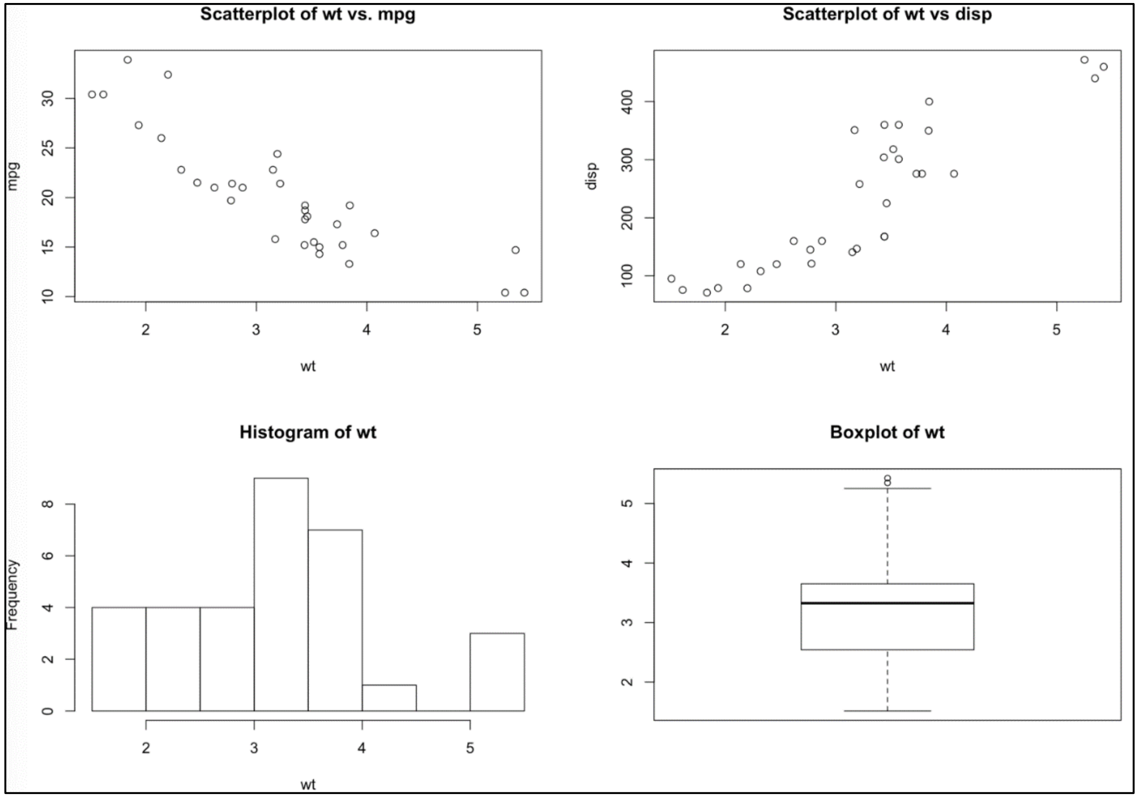
- Combining Plots
  - **R** makes it easy to combine multiple plots into one overall graph, using either the **`par( )`** or **`layout( )`** function.
  - With the **`par()`** function, you can include the option **`mfrow=c(nrows, ncols)`** to create a matrix of *nrows x ncols* plots that are filled in by **`row. mfcol=c(nrows, ncols)`** fills in the matrix by columns.

Plotting Arguments

Example 5.13: Combining Plots

```
attach(mtcars)
par(mfrow=c(2,2))
plot(wt,mpg, main="Scatterplot of wt vs. mpg")
plot(wt,disp, main="Scatterplot of wt vs disp")
hist(wt, main="Histogram of wt")
boxplot(wt, main="Boxplot of wt")
```

Plotting Arguments



Outline for this Session

- Plotting Commands
- Specialized Plotting Commands
- Plotting Arguments
- **ggplot**

ggplot2

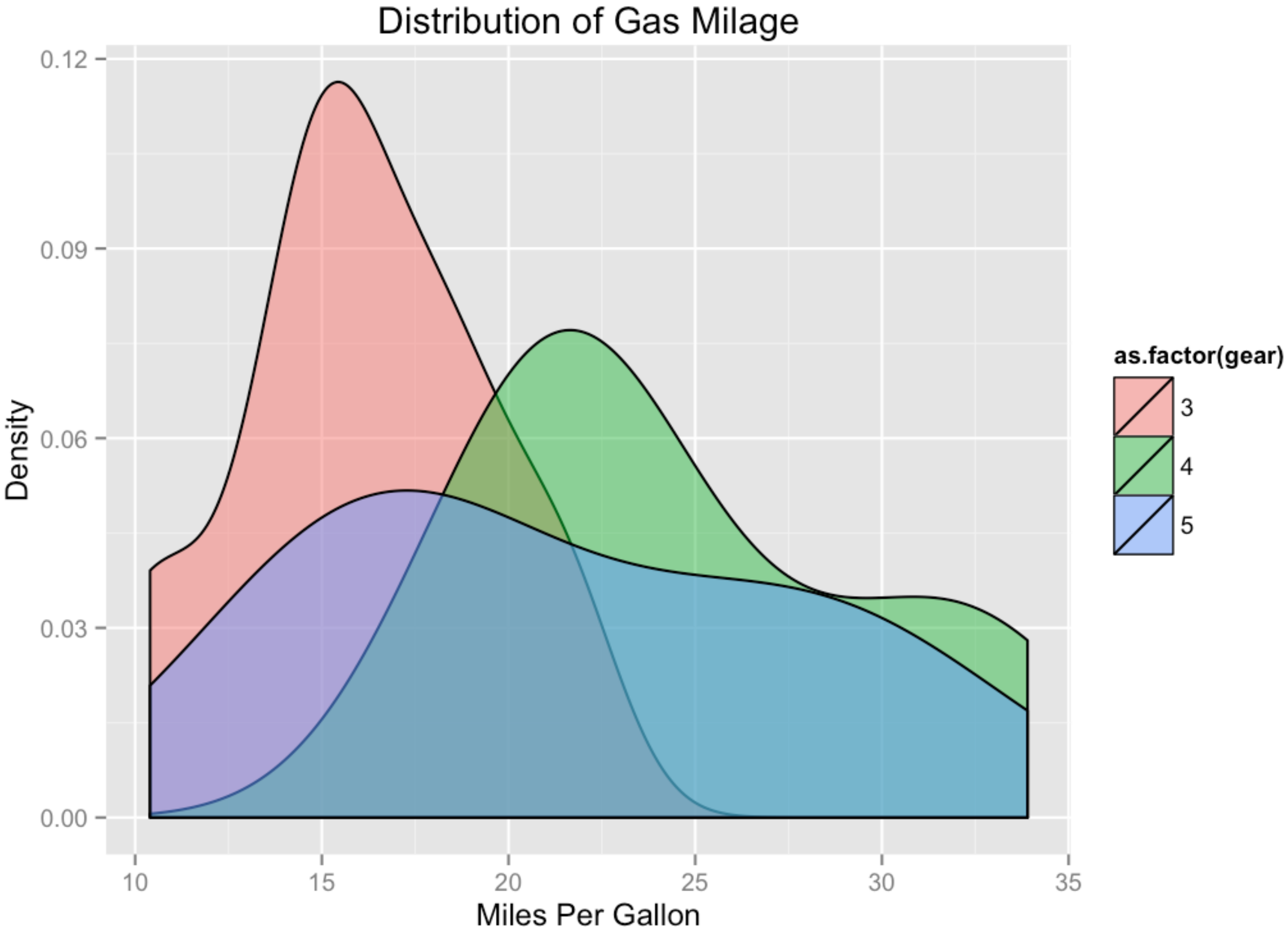
- The ggplot2 package, created by Hadley Wickham, offers a powerful graphics language for creating elegant and complex plots. Its popularity in the R community has exploded in recent years.
- Originally based on Leland Wilkinson's The Grammar of Graphics, ggplot2 allows you to create graphs that represent both univariate and multivariate numerical and categorical data in a straightforward manner.

ggplot2

Example 5.14: Kernel Density Plots

```
#Kernel density plots for mpg
#grouped by number of gears (indicated by color)
library("ggplot2")
qplot(mpg, data=mtcars, geom="density",
fill=as.factor(gear), alpha=I(.5),
main="Distribution of Gas Mileage",
xlab="Miles Per Gallon",
ylab="Density")
```

ggplot2


Distribution of Gas Milage
Density
0.12
0.09
0.06
0.03
0.00
10
15
20
25
30
35
Miles Per Gallon
as.factor(gear)
3
4
5

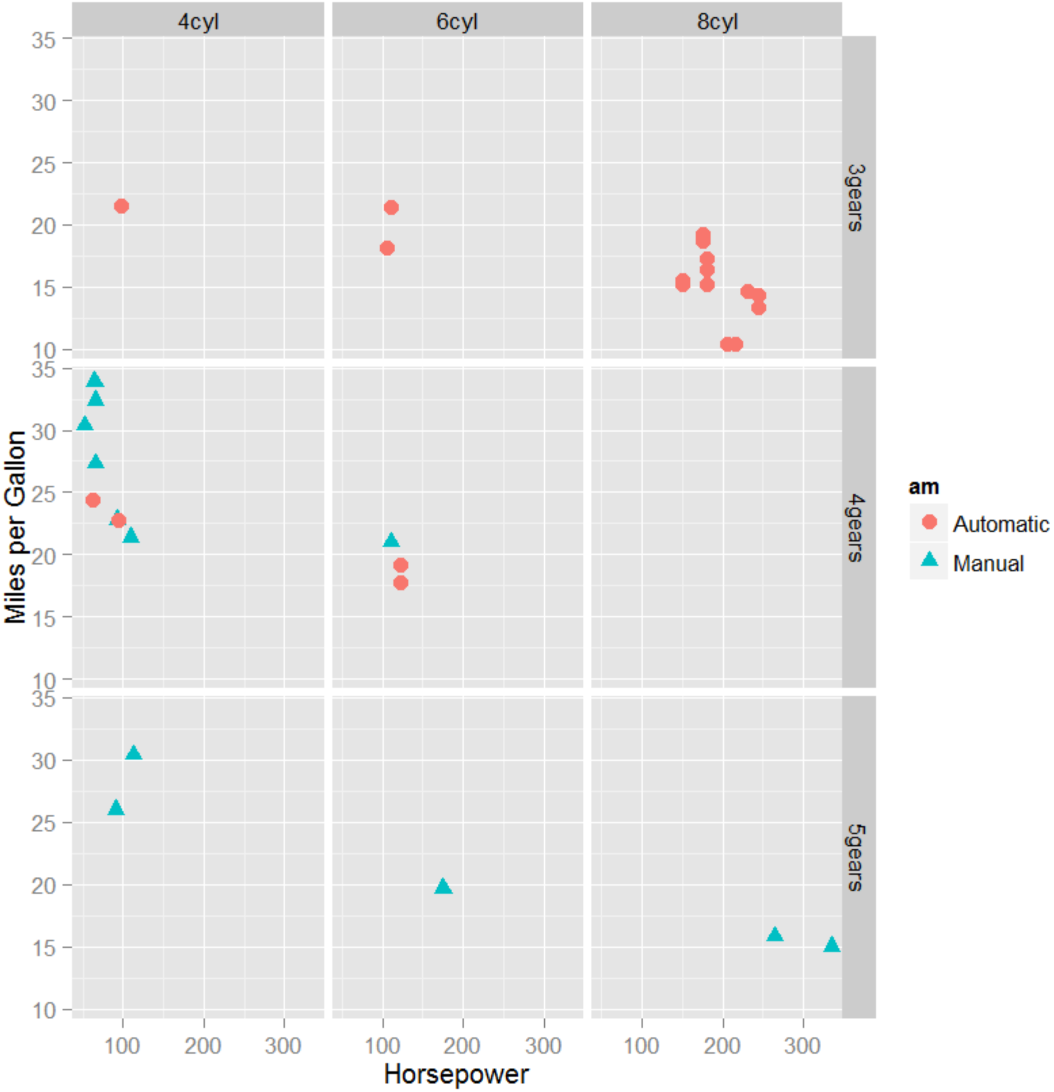
ggplot2

Example 5.15: Scatterplot using ggplot

- #Scatterplot of mpg vs. hp for each combination of gears and cylinders
- #in each facet, transmission type is represented by shape and color

-
  ```
  qplot(hp, mpg, data=mtcars,
  shape=as.factor(am),
  color=am,facets=gear~cyl, size=I(3),
  xlab="Horsepower", ylab="Miles per
  Gallon")
  ```

ggplot2


4cyl
6cyl
8cyl
3gears
4gears
5gears
35
30
25
20
15
10
100
200
300
Miles per Gallon
Horsepower
am
● Automatic
▲ Manual

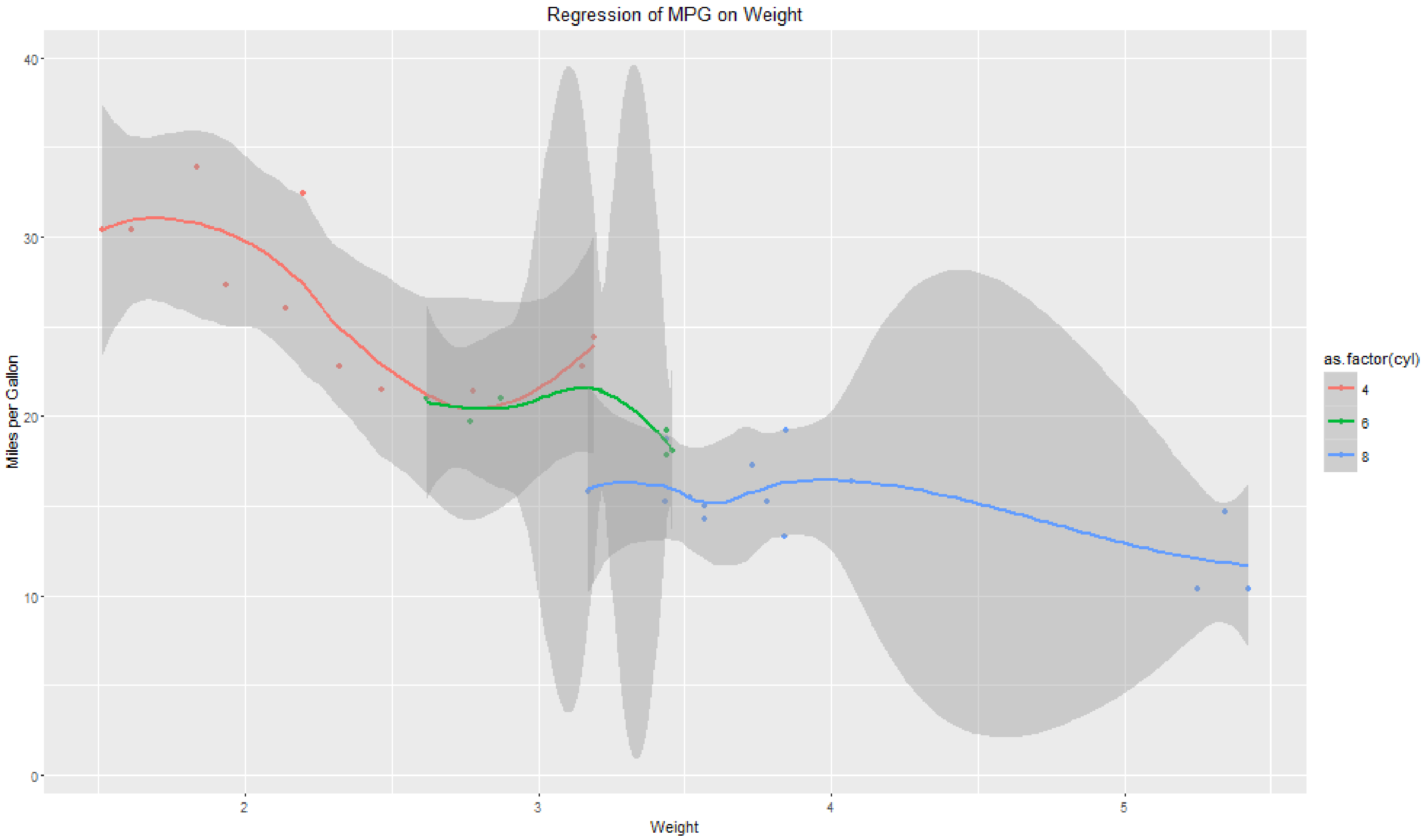
ggplot2

Example 5.16: Line Plots

```
# Separate regressions of mpg on weight for each number of cylinders
```

```
qplot(wt, mpg, data=mtcars, geom=c("point", "smooth"), color=as.factor(cyl), main="Regression of MPG on Weight", xlab="Weight", ylab="Miles per Gallon")
```

ggplot2



Regression of MPG on Weight
Miles per Gallon
40
30
20
10
0
Weight
2
3
4
5
as.factor(cyl)
4
6
8

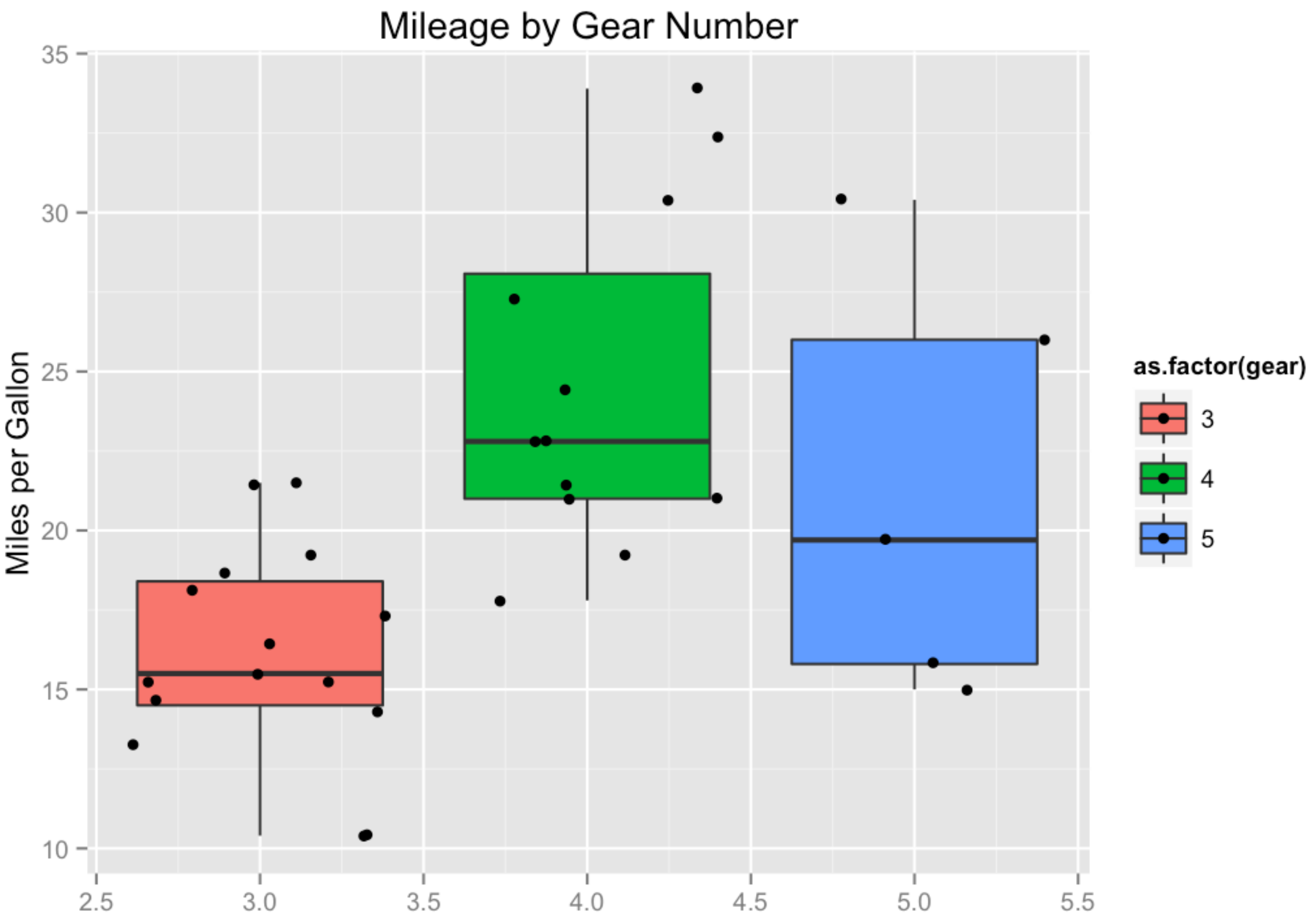
ggplot2

Example 5.17: Box Plots

- #Boxplots of mpg by number of gears
- #observations (points) are overlayed and jittered

-
  ```
  qplot(gear, mpg, data=mtcars,
  geom=c("boxplot", "jitter"),
  fill=as.factor(gear), main="Mileage by
  Gear Number",xlab="", ylab="Miles per
  Gallon")
  ```

ggplot2


Mileage by Gear Number
Miles per Gallon
35
30
25
20
15
10
2.5
3.0
3.5
4.0
4.5
5.0
5.5
as.factor(gear)
3
4
5

Outline for this Session

- Plotting Commands
- Specialized Plotting Commands
- Plotting Arguments
- ggplot

References

- http://www.statmethods.net/advgraphs/parameters.html
- http://www.statmethods.net/advgraphs/ggplot2.html