Stat 5810, Section 003
Statistical Visualization I
Fall 2018

Dr. Jürgen Symanzik

Utah State University

Department of Mathematics and Statistics

3900 Old Main Hill

Logan, UT 84322–3900

Tel.: (435) 797–0696

FAX: (435) 797–1822

e-mail: `symanzik@math.usu.edu`

Web: `http://www.math.usu.edu/~symanzik/`

Scans//PhdcomicsCom_Plotting.jpg

Figure 1: http://www.phdcomics.com/comics/archive.php?comicid=1541, Cartoon.

Contents

Acknowledgements

This course uses some of the course materials provided by Dr. Mike Minnotte (formerly USU, now with the University of North Dakota) as held in the Fall 2006 semester. Additional materials have been taken from other Statistical Graphics courses, such as the ones offered by Dr. Di Cook (formerly Iowa State University; now Monash University: `http://dicook.org/`) and Dr. Dan Carr (George Mason University: `http://mason.gmu.edu/~dcarr/`). Other examples and R code originate from Heike Hofmann, Paul Murrell, Carson Sievert, Martin Theus, Antony Unwin, Simon Urbanek, Hadley Wickham, Lee Wilkinson, and others. We are likely to include parts from additional authors and sources that will be specified later during the semester.

Thanks are also due to 60+ students and guests who took the former "Stat 6560: Graphical Methods" and the current "Statistical Visualization I & II" courses with me since the Spring 2009 semester for their valuable comments that helped to improve, correct, and extend these lecture notes.

Jürgen Symanzik, September 10, 2018.

`Scans//Zelazny_px_Fig.jpg`

Figure 2: **?**, p. x, Cartoon.

1 Introduction

1.1 Goals of the Course

The course answers three main questions:

- Q: Why statistical graphics (and which ones to draw)? —
  A: **?**, p. xi, indicates that statistical graphics can be used for
  - data cleaning
  - exploring the data structure
  - detecting outliers and unusual groups
  - identifying trends and clusters
  - spotting local patterns
  - evaluating modeling output
  - presenting results
  - exploratory data analysis (EDA)
  - data mining

  Different types of graphics answer different questions. Often, we draw a large number of (different) graphics to better understand the full data set.

- Q: How to construct statistical graphics in R? —
  A: Many R packages and supporting books exists, e.g.,
  - baseR, see **?**
  - `ggplot2`, see **?** and **?**
  - `lattice`, see **?** and **?**
  - general overview of R graphics, see **?**

  This course is not about a single R package, but rather introduces, uses, and compares a large number of R packages for various types of graphics.

- Q: How to distinguish between **good** and **bad** statistical graphics? —
  A: see the next section for a first brief answer

1.2 Motivation: Bad Graphics

Statistical graphics and data visualization are critical elements of modern data analysis and presentation. From initial exploration of a data set to the final presentation of results to the end user, statistical graphics play a vital role in shaping our understanding of our data. Through proper use of graphics, we can make critical discoveries, and communicate them clearly. Conversely, poor use or misuse of graphics can seriously mislead (by accident or design).

In the recent past, three examples that show the misuse of statistical graphics have been widely discussed — in the statistical community and beyond.

Example 1:

Scans//FoxNews_Bush-cuts2013.png

Figure 3: Figure taken from `http://flowingdata.com/2012/08/06/fox-news-continues-charting-excellence/` on 9/13/2017.

Based on this graphic (and without looking at the axis labels and percentages) how much higher would the top tax rate be if the Bush tax cuts were to expire at the end of 2012? Really ?!?!?

Scans//FoxNews_Bush-cuts2013_corrected.png

Figure 4: Corrected figure taken from `http://flowingdata.com/2012/08/06/fox-news-continues-charting-excellence/` on 9/13/2017.

Example 2:

Scans//Obamacare1.pdf

Figure 5: Figure taken from https://www.mediamatters.org/blog/2014/03/31/dishonest-fox-charts-obamacare-enrollment-editi/198679 on 3/31/2014.

Based on this graphic (and without looking at the numbers in the graphic) how much of the target Obamacare enrollment has been reached 4 days prior to the March 31 deadline? Really ?!?!?

Scans//Obamacare2.jpg

Figure 6: Corrected figure taken from **?**, Figure 10, showing the “tip of the iceberg” of the bar chart.

Example 3:

What is Planned Parenthood mostly doing in 2013, according to Rep. Jason Chaffetz, R-Utah? Really ?!?!?

Scans//PlannedParenthood1.jpg

Figure 7: Rep. Jason Chaffetz, R-Utah, projected this chart during a high-profile congressional hearing investigating Planned Parenthood. Figure taken from `http://www.politifact.com/truth-o-meter/statements/2015/oct/01/jason-chaffetz/chart-shown-planned-parenthood-hearing-misleading-/` on 10/12/2017.

Politifact concluded:

> "At the hearing, Chaffetz presented a chart that showed the number of abortions at Planned Parenthood rising higher than the number of preventive services and cancer screenings between 2006 and 2013.
>
> But that's inaccurate, disputed by the chart's own, hard to read numerical labels. In fact, there were three times as many cancer screenings and prevention services as abortions in 2013. Experts in data presentation said this was an egregious example of using a chart to mislead.
>
> We rate the claim Pants on Fire."

Scans//PlannedParenthood2.jpg

Figure 8: Figure taken from `http://www.politifact.com/truth-o-meter/statements/2015/oct/01/jason-chaffetz/chart-shown-planned-parenthood-hearing-misleading-/` on 10/12/2017.

1.3 Motivation: Why Graphics ?!?

Why do we need graphics at all. Aren't summary statistics sufficient?

Start R and load the **?** data sets. Just type `anscombe` and take a first glance at the data sets. What do you notice?

```
> anscombe

   x1 x2 x3 x4    y1   y2    y3    y4
1  10 10 10  8  8.04 9.14  7.46  6.58
```

Scans//PlannedParenthood3.png

Figure 9: Corrected figure taken from http://www.politifact.com/truth-o-meter/statements/2015/oct/01/jason-chaffetz/chart-shown-planned-parenthood-hearing-misleading-/ on 10/12/2017.

2	8	8	8	8	6.95	8.14	6.77	5.76
3	13	13	13	8	7.58	8.74	12.74	7.71
4	9	9	9	8	8.81	8.77	7.11	8.84
5	11	11	11	8	8.33	9.26	7.81	8.47
6	14	14	14	8	9.96	8.10	8.84	7.04
7	6	6	6	8	7.24	6.13	6.08	5.25
8	4	4	4	19	4.26	3.10	5.39	12.50
9	12	12	12	8	10.84	9.13	8.15	5.56
10	7	7	7	8	4.82	7.26	6.42	7.91
11	5	5	5	8	5.68	4.74	5.73	6.89

Then calculate some summary statistics (separately for the four columns of X's and Y's): mean of the X's, mean of the Y's, standard deviation of the X's, standard deviation of the Y's, correlation coefficient, slope and intercept of the regression line, rms error.

```
> # calculate some summary statistics (separately for the
> # four columns of X's and Y's)
>
> # mean of the X's
> mean(anscombe$x1)

[1] 9

> mean(anscombe$x2)

[1] 9

> mean(anscombe$x3)

[1] 9

> mean(anscombe$x4)

[1] 9

> # mean of the Y's
> mean(anscombe$y1)

[1] 7.500909

> mean(anscombe$y2)

[1] 7.500909

> mean(anscombe$y3)

[1] 7.5
```

```
> mean(anscombe$y4)

[1] 7.500909

> # standard deviation of the X's
> sqrt(var(anscombe$x1))

[1] 3.316625

> sqrt(var(anscombe$x2))

[1] 3.316625

> sqrt(var(anscombe$x3))

[1] 3.316625

> sqrt(var(anscombe$x4))

[1] 3.316625

> # standard deviation of the Y's
> sqrt(var(anscombe$y1))

[1] 2.031568

> sqrt(var(anscombe$y2))

[1] 2.031657

> sqrt(var(anscombe$y3))

[1] 2.030424

> sqrt(var(anscombe$y4))

[1] 2.030579
```

```r
> # correlation coefficient
> cor(anscombe$x1, anscombe$y1)

[1] 0.8164205

> cor(anscombe$x2, anscombe$y2)

[1] 0.8162365

> cor(anscombe$x3, anscombe$y3)

[1] 0.8162867

> cor(anscombe$x4, anscombe$y4)

[1] 0.8165214

> # slope of the regression line
> slope1 <- cor(anscombe$x1, anscombe$y1) * sqrt(var(anscombe$y1)) /
+   sqrt(var(anscombe$x1))
> slope2 <- cor(anscombe$x2, anscombe$y2) * sqrt(var(anscombe$y2)) /
+   sqrt(var(anscombe$x2))
> slope3 <- cor(anscombe$x3, anscombe$y3) * sqrt(var(anscombe$y3)) /
+   sqrt(var(anscombe$x3))
> slope4 <- cor(anscombe$x4, anscombe$y4) * sqrt(var(anscombe$y4)) /
+   sqrt(var(anscombe$x4))
> slope1

[1] 0.5000909

> slope2

[1] 0.5

> slope3

[1] 0.4997273
```

```
> slope4

[1] 0.4999091

> # intercept of the regression line
> intercept1 <- mean(anscombe$y1) - slope1 * mean(anscombe$x1)
> intercept2 <- mean(anscombe$y2) - slope2 * mean(anscombe$x2)
> intercept3 <- mean(anscombe$y3) - slope3 * mean(anscombe$x3)
> intercept4 <- mean(anscombe$y4) - slope4 * mean(anscombe$x4)
> intercept1

[1] 3.000091

> intercept2

[1] 3.000909

> intercept3

[1] 3.002455

> intercept4

[1] 3.001727

> # rms error
> rmserror1 <- sqrt(1 - cor(anscombe$x1, anscombe$y1)^2) * sqrt(var(anscombe$y1))
> rmserror2 <- sqrt(1 - cor(anscombe$x2, anscombe$y2)^2) * sqrt(var(anscombe$y2))
> rmserror3 <- sqrt(1 - cor(anscombe$x3, anscombe$y3)^2) * sqrt(var(anscombe$y3))
> rmserror4 <- sqrt(1 - cor(anscombe$x4, anscombe$y4)^2) * sqrt(var(anscombe$y4))
> rmserror1

[1] 1.173145

> rmserror2

[1] 1.173724
```

```
> rmserror3
```

```
[1] 1.172868
```

```
> rmserror4
```

```
[1] 1.172284
```

So, the four pairs of X/Y columns basically are identical !?!

But, didn't we forget to **plot** the data !!!

```
> # based on: http://pbil.univ-lyon1.fr/library/base/html/anscombe.html
> # extracted and adapted on 1/6/09
>
> ##-- now some "magic" to do the 4 regressions in a loop:
> ff <- y ~ x
> class(ff)
```

```
[1] "formula"
```

```
> ff[1]
```

```
`~`()
```

```
> ff[2]
```

```
y()
```

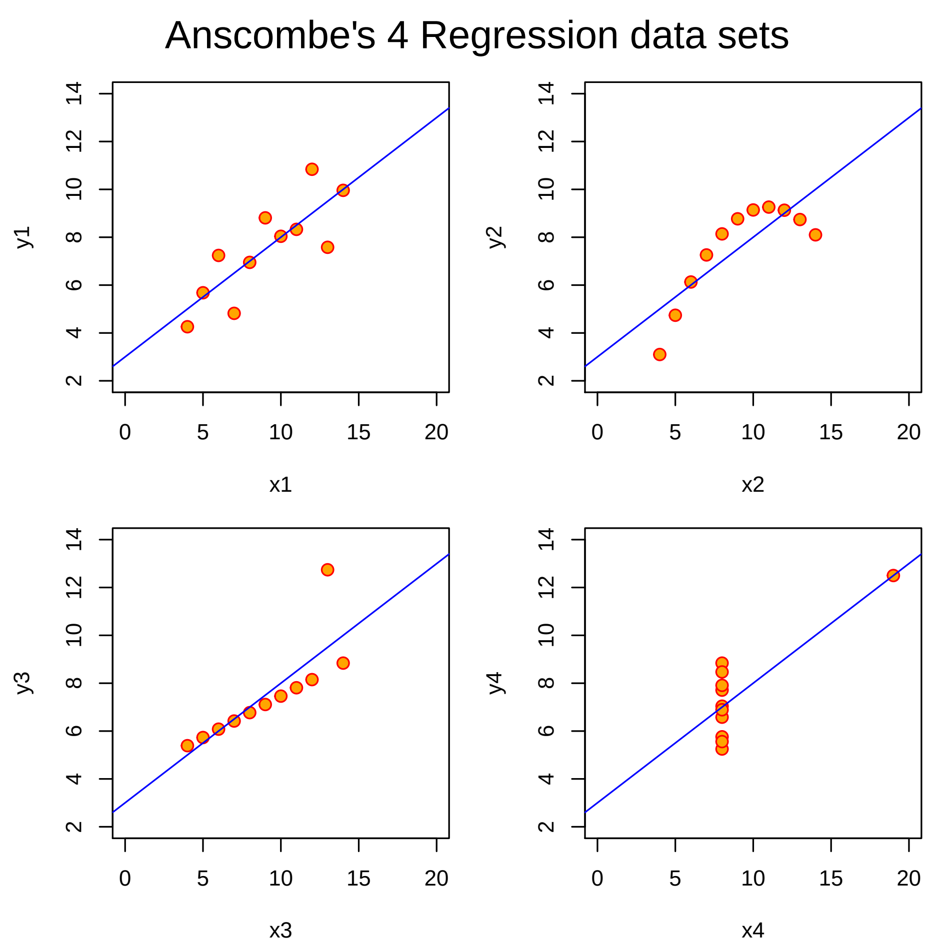
```
> ff[3]
```

```
x()
```

```
> for (i in 1:4)
+ {
+   ff[2:3] <- lapply(paste(c("y", "x"), i, sep = ""), as.name)
+   assign(paste("lm.", i, sep = ""), lmi <- lm(ff, data = anscombe))
+ }
> lm.1
```

```
Call:
lm(formula = ff, data = anscombe)

Coefficients:
(Intercept)           x1
     3.0001       0.5001

> ## See how close they are (numerically!)
> sapply(objects(pattern = "lm.[1-4]"), function(n) coef(get(n)))

                 lm.1     lm.2      lm.3      lm.4
(Intercept) 3.0000909 3.000909 3.0024545 3.0017273
x1          0.5000909 0.500000 0.4997273 0.4999091

> lapply(objects(pattern = "lm.[1-4]"), function(n) summary(get(n))$coef)

[[1]]
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.0000909  1.1247468 2.667348 0.025734051
x1          0.5000909  0.1179055 4.241455 0.002169629

[[2]]
            Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.000909  1.1253024 2.666758 0.025758941
x2          0.500000  0.1179637 4.238590 0.002178816

[[3]]
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.0024545  1.1244812 2.670080 0.025619109
x3          0.4997273  0.1178777 4.239372 0.002176305

[[4]]
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 3.0017273  1.1239211 2.670763 0.025590425
x4          0.4999091  0.1178189 4.243028 0.002164602
```

```
> ## Now, do what you should have done in the first place: PLOTS
> op <- par(mfrow = c(2, 2), mar = .1 + c(4, 4, 1, 1), oma = c(0, 0, 2, 0))
> for (i in 1:4)
+ {
+   ff[2:3] <- lapply(paste(c("y", "x"), i, sep = ""), as.name)
+   plot(ff, data = anscombe, col = "red", pch = 21, bg = "orange",
+         cex = 1.2, xlim = c(0, 20), ylim = c(2, 14))
+   abline(get(paste("lm.", i, sep = "")), col = "blue")
+ }
> mtext("Anscombe's 4 Regression data sets", outer = TRUE, cex = 1.5)
> par(op)
```



See here for additional references:

`http://en.wikipedia.org/wiki/Anscombe's_quartet`

?, p. 13, concludes:

> “Graphics *reveal* data. Indeed graphics can be more precise and revealing than conventional statistical computations. Consider Anscombe’s quartet: all four of these data sets are described by exactly the same linear model (at least until the residuals are examined).”

The Anscombe data show up in numerous textbooks, as early as in **?**, pp. 131–134, and as recent as in **?**, p. 120 (Exercise 2.73). In fact, this data set should be shown in every undergraduate class as well as in every regression class to demonstrate what might happen when blindly performing any statistical calculations without plotting the data first.

1.4 Further Reading

In addition to **?** cited so far in this chapter, many other sources exist that make a strong case why to use graphics. Some of these additional sources are:

- **?**
- **?**

`Scans//AmstatNews_Jan2009_p25_Fig.jpg`

Figure 10: Amstat News, January 2009, p. 25, Cartoon.

2 Basic Graph Construction and Refinement

(Based on ?, Chapter 1: Setting the Scene

The supporting materials for **?** can be obtained from `http://www.gradaanwr.net/`. In particular, the original R code for each chapter can be downloaded as a zip file from `http://www.gradaanwr.net/content/`. We will work with modified versions of some of the provided code in class.

2.1 Figure 1.1

```
World Speed Skiing Competition, Verbier 21st April, 2011

Description
There were separate Speed Skiing competitions for men (79 participants) and
women (12 participants).

Usage
data(SpeedSki)
```

```
> ##Libraries
> library(ggplot2)
> library(gridExtra)
> library(ggthemes)
> library(dplyr)
> library(GGally)
> library(vcd)
> library(extracat)
> library(GDAdata)
> library(plotly)
> ##Settings
> palette("default")
> update_geom_defaults("bar", list(fill = "grey70", colour = "grey40"))
> scale_colour_discrete <- function(...) scale_colour_brewer(..., palette = "Set2")
> scale_fill_discrete <- function(...) scale_fill_colorblind()
> auTheme <- theme_grey()  +
+   theme(panel.background = element_rect(colour = NA, fill = "grey90")) +
```

```
+   theme(plot.background = element_rect(colour = NA, fill = "grey90")) +
+   theme(legend.background = element_rect(fill = "grey90")) +
+   theme(plot.title = element_text(vjust = 2))
> theme_set(auTheme)
> ## ----speedski---- Fig 1.1
> data(SpeedSki, package = "GDAdata")
> # step-by-step
>
> # basic first graphic
> ggplot(SpeedSki, aes(x = Speed)) +
+   geom_histogram()
> # adjust range of x-axis
> summary(SpeedSki$Speed)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  160.2   171.8   183.1   184.1   192.3   211.7

> ggplot(SpeedSki, aes(x = Speed)) +
+   xlim(160, 220) +
+   geom_histogram()
> # adjust binwidth
> ggplot(SpeedSki, aes(x = Speed)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5)
> # add axis labels
> ggplot(SpeedSki, aes(x = Speed)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5) +
+   xlab("Speed (km/hr)") +
+   ylab("")
> # condition on gender in 2 related histograms (small multiples!)
> ggplot(SpeedSki, aes(x = Speed)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
```

```
+   facet_wrap(~ Sex, ncol = 1)
> # use color as a distinction for gender
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1)
> # omit legend = code from the book
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> # final result in book
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> # further adjust y-axis range
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   ylim(0, 12) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> # further adjust y-axis ticks & gridlines
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
```

```
+   ylim(0, 12) +
+   scale_y_continuous(breaks = seq(0, 12, 2)) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> # interactive version
> ggplotly()
> # save as external jpg file
> jpeg("Speedski.jpg")
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   ylim(0, 12) +
+   scale_y_continuous(breaks = seq(0, 12, 2)) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> dev.off()
```

```
pdf
  2
```

```
> # save as external pdf file
> pdf("Speedski.pdf")
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   ylim(0, 12) +
+   scale_y_continuous(breaks = seq(0, 12, 2)) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> dev.off()
```

```
pdf
  2
```

```
> # Question: How to extract the intervals from a ggplot histogram object?
> # Answer based on: https://stackoverflow.com/questions/25378184/need-to-extract-data
>
> # assign ggplot object to a variable
> g <- ggplot(SpeedSki, aes(x = Speed)) +
+   geom_histogram()
> class(g)
```

```
[1] "gg"     "ggplot"
```

```
> g
> # extract plot information
> pg <- ggplot_build(g)
> class(pg)
```

```
[1] "ggplot_built"
```

```
> names(pg)
```

```
[1] "data"   "layout" "plot"
```

```
> # look at the data part
> head(pg$data[[1]])
```

```
  y count        x     xmin     xmax  density    ncount  ndensity PANEL group
1 1     1 159.6724 158.7853 160.5595 0.006194 0.1111111 0.1111111     1    -1
2 1     1 161.4466 160.5595 162.3336 0.006194 0.1111111 0.1111111     1    -1
3 1     1 163.2207 162.3336 164.1078 0.006194 0.1111111 0.1111111     1    -1
4 4     4 164.9948 164.1078 165.8819 0.024776 0.4444444 0.4444444     1    -1
5 5     5 166.7690 165.8819 167.6560 0.030970 0.5555556 0.5555556     1    -1
6 6     6 168.5431 167.6560 169.4302 0.037164 0.6666667 0.6666667     1    -1
  ymin ymax   fill colour size linetype alpha
1    0    1 grey70 grey40  0.5        1    NA
```

```
2    0    1 grey70 grey40  0.5        1    NA
3    0    1 grey70 grey40  0.5        1    NA
4    0    4 grey70 grey40  0.5        1    NA
5    0    5 grey70 grey40  0.5        1    NA
6    0    6 grey70 grey40  0.5        1    NA
```

```
> # compare with the modified intervals
>
> g2 <- ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> g2
> pg2 <- ggplot_build(g2)
> head(pg2$data[[1]])
```

```
     fill y count      x  xmin  xmax    density    ncount  ndensity PANEL group
1 #000000 3     3 161.25 160.0 162.5 0.10000000 1.0000000 1.0000000     1     1
2 #000000 0     0 163.75 162.5 165.0 0.00000000 0.0000000 0.0000000     1     1
3 #000000 1     1 166.25 165.0 167.5 0.03333333 0.3333333 0.3333333     1     1
4 #000000 1     1 168.75 167.5 170.0 0.03333333 0.3333333 0.3333333     1     1
5 #000000 0     0 171.25 170.0 172.5 0.00000000 0.0000000 0.0000000     1     1
6 #000000 0     0 173.75 172.5 175.0 0.00000000 0.0000000 0.0000000     1     1
  ymin ymax colour size linetype alpha
1    0    3 grey40  0.5        1    NA
2    0    0 grey40  0.5        1    NA
3    0    1 grey40  0.5        1    NA
4    0    1 grey40  0.5        1    NA
5    0    0 grey40  0.5        1    NA
6    0    0 grey40  0.5        1    NA
```

```
> ## BaseR
>
```

```
> # basic first graphic
> hist(SpeedSki$Speed)
> # adjust range of x-axis & binwidth
> hist(SpeedSki$Speed,
+      breaks = seq(160, 220, by = 2.5))
> # add axis labels
> hist(SpeedSki$Speed,
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "")
> # condition on gender in 2 related histograms (small multiples!)
> hist(SpeedSki$Speed[SpeedSki$Sex == "Female"],
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "")
> hist(SpeedSki$Speed[SpeedSki$Sex == "Male"],
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "")
> # combine into 1 figure
> op <- par(no.readonly = TRUE) # save original graphical parameters
> par(mfrow = c(2, 1))
> hist(SpeedSki$Speed[SpeedSki$Sex == "Female"],
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "")
> hist(SpeedSki$Speed[SpeedSki$Sex == "Male"],
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "")
> # add individual main titles
> par(mfrow = c(2, 1))
> hist(SpeedSki$Speed[SpeedSki$Sex == "Female"],
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "",
```

```
+      main = "Female")
> hist(SpeedSki$Speed[SpeedSki$Sex == "Male"],
+      breaks = seq(160, 220, by = 2.5),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Male")
> # adjust y-axis to common scale (small multiples!)
> par(mfrow = c(2, 1))
> hist(SpeedSki$Speed[SpeedSki$Sex == "Female"],
+      breaks = seq(160, 220, by = 2.5),
+      ylim = c(0, 12),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Female")
> hist(SpeedSki$Speed[SpeedSki$Sex == "Male"],
+      breaks = seq(160, 220, by = 2.5),
+      ylim = c(0, 12),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Male")
> # reduce outer margins # c(bottom, left, top, right)
> par(mfrow = c(2, 1),
+     oma = c(0, 0, 0, 0))
> hist(SpeedSki$Speed[SpeedSki$Sex == "Female"],
+      breaks = seq(160, 220, by = 2.5),
+      ylim = c(0, 12),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Female")
> hist(SpeedSki$Speed[SpeedSki$Sex == "Male"],
+      breaks = seq(160, 220, by = 2.5),
+      ylim = c(0, 12),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Male")
> # reduce inner margins # c(bottom, left, top, right)
```

```
> par(mfrow = c(2, 1),
+     oma = c(0, 0, 0, 0),
+     mar = c(5, 3, 1, 0))
> hist(SpeedSki$Speed[SpeedSki$Sex == "Female"],
+      breaks = seq(160, 220, by = 2.5),
+      ylim = c(0, 12),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Female")
> hist(SpeedSki$Speed[SpeedSki$Sex == "Male"],
+      breaks = seq(160, 220, by = 2.5),
+      ylim = c(0, 12),
+      xlab = "Speed (km/hr)",
+      ylab = "",
+      main = "Male")
> par(op) # reset par for future graphics
```

2.2 Figure 1.2

```
> ## ----speedski2---- Fig 1.2
>
> # final result in book for Fig 1.1
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_wrap(~ Sex, ncol = 1) +
+   theme(legend.position = "none")
> # different layout
> ggplot(SpeedSki, aes(x = Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25) +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_grid(~ Sex) +
```

```
+   theme(legend.position = "none")
> # condition on event
> ggplot(SpeedSki, aes(Speed, fill = Sex)) +
+   geom_histogram(binwidth = 2.5)  +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_grid(Sex ~ Event) +
+   theme(legend.position = "none")
> # readjust range and center of bins
> ggplot(SpeedSki, aes(Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25)  +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_grid(Sex ~ Event) +
+   theme(legend.position = "none")
> # interactive version
> ggplotly()
> #try a few things yourself!
> names(SpeedSki)

 [1] "Rank"       "Bib"        "FIS.Code"   "Name"       "Year"
 [6] "Nation"     "Speed"      "Sex"        "Event"      "no.of.runs"

> head(SpeedSki)

  Rank Bib FIS.Code                  Name Year Nation  Speed  Sex     Event
1    1  61     7039        ORIGONE Simone 1979    ITA 211.67 Male Speed One
2    2  59     7078          ORIGONE Ivan 1987    ITA 209.70 Male Speed One
3    3  66   190130        MONTES Bastien 1985    FRA 209.69 Male Speed One
4    4  57     7178 SCHROTTSHAMMER Klaus 1979    AUT 209.67 Male Speed One
5    5  69   510089          MAY Philippe 1970    SUI 209.19 Male Speed One
6    6  75     7204           BILLY Louis 1993    FRA 208.33 Male Speed One
  no.of.runs
1          4
2          4
```

```
3          4
4          4
5          4
6          4

> ggplot(SpeedSki, aes(Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25)  +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_grid(Sex ~ no.of.runs ~ Event) +
+   theme(legend.position = "none")
> ggplot(SpeedSki, aes(Speed, fill = Sex)) +
+   xlim(160, 220) +
+   geom_histogram(binwidth = 2.5, center = 1.25)  +
+   xlab("Speed (km/hr)") +
+   ylab("") +
+   facet_grid(Sex ~ Nation ~ Event) +
+   theme(legend.position = "none")
```

2.3 Figure 1.3

```
Edgar Anderson's Iris Data

Description
This famous (Fisher's or Anderson's) iris data set gives the measurements
in centimeters of the variables sepal length and width and petal length and width,
respectively, for 50 flowers from each of 3 species of iris. The species are
Iris setosa, versicolor, and virginica.

Usage
iris
iris3

> ## ----petal1---- Fig 1.3
>
```

```
> # basic first graphic = code from book
> ggplot(iris, aes(Petal.Length)) +
+   geom_histogram()
> # adjust binwidth & center
> ggplot(iris, aes(Petal.Length)) +
+   geom_histogram(binwidth = 0.5, center = 0.25)
> # adjust xlim & ylim
> ggplot(iris, aes(Petal.Length)) +
+   xlim(0, 8) +
+   ylim(0, 40) +
+   geom_histogram(binwidth = 0.5, center = 0.25)
> # interesting: notice the following
> summary(iris$Petal.Length)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.600   4.350   3.758   5.100   6.900

> # interactive version
> ggplotly()
> ## BaseR
>
> # basic first graphic
> hist(iris$Petal.Length)
> # add axis label & title
> hist(iris$Petal.Length,
+      xlab = "Petal Length",
+      main = "Iris Data Set")
> # adjust y-axis range
> hist(iris$Petal.Length,
+      ylim = c(0, 40),
+      xlab = "Petal Length",
+      main = "Iris Data Set")
> # adjust starting interval for x-axis
> hist(iris$Petal.Length,
+      breaks = seq(0, 7, by = 0.5),
+      ylim = c(0, 40),
```

```
+      xlab = "Petal Length",
+      main = "Iris Data Set")
```

2.4 Figure 1.4

```
> ## ----scpetal---- Fig 1.4
>
> # basic first graphic
> ggplot(iris, aes(Petal.Length, Petal.Width)) +
+   geom_point()
> # add color to distinguish species
> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) +
+   geom_point()
> # place legend at bottom
> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) +
+   geom_point() +
+   theme(legend.position = "bottom")
> # choose a different color scheme for colorblind viewers = code from book
> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) +
+   geom_point() +
+   theme(legend.position = "bottom") +
+   scale_colour_colorblind()
> # interactive version
> ggplotly()
> # try a few things
> ggplot(iris, aes(Petal.Length, Petal.Width, shape = Species)) +
+   geom_point() +
+   theme(legend.position = "bottom")
> ggplot(iris, aes(Petal.Length, Petal.Width, shape = Species, color = Species)) +
+   geom_point() +
+   theme(legend.position = "bottom")
> ggplot(iris, aes(Petal.Length, Petal.Width, shape = Species, color = Species, size =
+   geom_point() +
+   theme(legend.position = "bottom")
> ## BaseR
>
```

```
> # basic plot
> plot(iris$Petal.Length, iris$Petal.Width)
> # add color to distinguish species
> plot(iris$Petal.Length, iris$Petal.Width,
+      col = iris$Species)
> # add axis labels & title
> plot(iris$Petal.Length, iris$Petal.Width,
+      col = iris$Species,
+      xlab = "Petal Length",
+      ylab = "Petal Width",
+      main = "Iris Data")
```

2.5 Figure 1.5

```
Student Admissions at UC Berkeley

Description
Aggregate data on applicants to graduate school at Berkeley for the six largest
departments in 1973 classified by admission and sex.

Usage
UCBAdmissions
```

```
> ## ----ucbaDeptx---- Fig 1.5
>
> # basic first graphics
> ucba <- as.data.frame(UCBAdmissions)
> a <- ggplot(ucba, aes(Dept)) +
+   geom_bar(aes(weight = Freq))
> b <- ggplot(ucba, aes(Gender)) +
+   geom_bar(aes(weight = Freq))
> c <- ggplot(ucba, aes(Admit)) +
+   geom_bar(aes(weight = Freq))
> a
> b
> c
```

```
> # arrange layout
> ucba <- as.data.frame(UCBAdmissions)
> a <- ggplot(ucba, aes(Dept)) +
+   geom_bar(aes(weight = Freq))
> b <- ggplot(ucba, aes(Gender)) +
+   geom_bar(aes(weight = Freq))
> c <- ggplot(ucba, aes(Admit)) +
+   geom_bar(aes(weight = Freq))
> grid.arrange(a, b, c)
> # refine layout
> ucba <- as.data.frame(UCBAdmissions)
> a <- ggplot(ucba, aes(Dept)) +
+   geom_bar(aes(weight = Freq))
> b <- ggplot(ucba, aes(Gender)) +
+   geom_bar(aes(weight = Freq))
> c <- ggplot(ucba, aes(Admit)) +
+   geom_bar(aes(weight = Freq))
> grid.arrange(a, b, c, nrow = 1)
> # adjust widths = code from book
> ucba <- as.data.frame(UCBAdmissions)
> a <- ggplot(ucba, aes(Dept)) +
+   geom_bar(aes(weight = Freq))
> b <- ggplot(ucba, aes(Gender)) +
+   geom_bar(aes(weight = Freq))
> c <- ggplot(ucba, aes(Admit)) +
+   geom_bar(aes(weight = Freq))
> grid.arrange(a, b, c, nrow = 1, widths = c(7, 3, 3))
```

2.6 Figure 1.6

```
> ## ----berkeleyS---- Fig 1.6
>
> # basic first graphic
> ucb <- data.frame(UCBAdmissions)
> doubledecker(xtabs(Freq ~ Dept + Gender + Admit, data = ucb))
> # modify colors
```

```
> ucb <- data.frame(UCBAdmissions)
> doubledecker(xtabs(Freq ~ Dept + Gender + Admit, data = ucb),
+              gp = gpar(fill = c("grey90", "steelblue")))
> # create new factor with differently sorted levels and swap top/bottom in bars
> ucb <- data.frame(UCBAdmissions)
> ucb <- within(ucb, Accept <-
+                 factor(Admit, levels = c("Rejected", "Admitted")))
> doubledecker(xtabs(Freq ~ Dept + Gender + Accept, data = ucb),
+              gp = gpar(fill = c("grey90", "steelblue")))
```

2.7 Figure 1.7

```
Diabetes in Pima Indian Women

Description
A population of women who were at least 21 years old, of Pima Indian heritage
and living near Phoenix, Arizona, was tested for diabetes according to
World Health Organization criteria. The data were collected by the
US National Institute of Diabetes and Digestive and Kidney Diseases.
We used the 532 complete records after dropping the (mainly missing) data on
serum insulin.

Usage
Pima.tr
Pima.tr2
Pima.te
```

```
> ## ----pimaHist---- Fig 1.7
>
> # basic first graphics
> data(Pima.tr2, package = "MASS")
> h1 <- ggplot(Pima.tr2, aes(glu)) +
+   geom_histogram()
> h2 <- ggplot(Pima.tr2, aes(bp)) +
+   geom_histogram()
> h3 <- ggplot(Pima.tr2, aes(skin)) +
```

```
+   geom_histogram()
> h4 <- ggplot(Pima.tr2, aes(bmi)) +
+   geom_histogram()
> h5 <- ggplot(Pima.tr2, aes(ped)) +
+   geom_histogram()
> h6 <- ggplot(Pima.tr2, aes(age)) +
+   geom_histogram()
> h1
> h2
> h3
> h4
> h5
> h6
> # arrange layout
> data(Pima.tr2, package = "MASS")
> h1 <- ggplot(Pima.tr2, aes(glu)) +
+   geom_histogram()
> h2 <- ggplot(Pima.tr2, aes(bp)) +
+   geom_histogram()
> h3 <- ggplot(Pima.tr2, aes(skin)) +
+   geom_histogram()
> h4 <- ggplot(Pima.tr2, aes(bmi)) +
+   geom_histogram()
> h5 <- ggplot(Pima.tr2, aes(ped)) +
+   geom_histogram()
> h6 <- ggplot(Pima.tr2, aes(age)) +
+   geom_histogram()
> grid.arrange(h1, h2, h3, h4, h5, h6)
> # refine layout = code from book
> data(Pima.tr2, package = "MASS")
> h1 <- ggplot(Pima.tr2, aes(glu)) +
+   geom_histogram()
> h2 <- ggplot(Pima.tr2, aes(bp)) +
+   geom_histogram()
> h3 <- ggplot(Pima.tr2, aes(skin)) +
+   geom_histogram()
```

```
> h4 <- ggplot(Pima.tr2, aes(bmi)) +
+   geom_histogram()
> h5 <- ggplot(Pima.tr2, aes(ped)) +
+   geom_histogram()
> h6 <- ggplot(Pima.tr2, aes(age)) +
+   geom_histogram()
> grid.arrange(h1, h2, h3, h4, h5, h6, nrow = 2)
```

2.8 Figure 1.8

```
> ## ----pimaBoxs---- Fig 1.8
>
> ## BaseR
>
> # basic first graphic
> PimaV <- select(Pima.tr2, glu:age)
> boxplot(PimaV)
> # use standardized scale
> PimaV <- select(Pima.tr2, glu:age)
> boxplot(scale(PimaV))
> # change symbol and color for outliers
> PimaV <- select(Pima.tr2, glu:age)
> boxplot(scale(PimaV), pch = 16, outcol = "red")
> # reduce margins = code from book
> PimaV <- select(Pima.tr2, glu:age)
> par(mar = c(3.1, 4.1, 1.1, 2.1))
> boxplot(scale(PimaV), pch = 16, outcol = "red")
> ## ggplot2
>
> # basic boxplot of glu (needs "var" as a dummy argument)
> PimaV <- select(Pima.tr2, glu:age)
> ggplot(PimaV, aes("var", glu)) +
+   geom_boxplot()
> # boxplot of glu with x-axis labels removed
> PimaV <- select(Pima.tr2, glu:age)
> ggplot(PimaV, aes("var", glu)) +
```

```
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> # all boxplots, arranged side-by-side
> PimaV <- select(Pima.tr2, glu:age)
> b1 <- ggplot(PimaV, aes("var", glu)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b2 <- ggplot(PimaV, aes("var", bp)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b3 <- ggplot(PimaV, aes("var", skin)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b4 <- ggplot(PimaV, aes("var", bmi)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b5 <- ggplot(PimaV, aes("var", ped)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b6 <- ggplot(PimaV, aes("var", age)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> grid.arrange(b1, b2, b3, b4, b5, b6, ncol = 6)
> # use standardized scale
> PimaV <- select(Pima.tr2, glu:age)
> PimaV <- as.data.frame(scale(PimaV))
> b1 <- ggplot(PimaV, aes("var", glu)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
```

```
+   geom_boxplot()
> b2 <- ggplot(PimaV, aes("var", bp)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b3 <- ggplot(PimaV, aes("var", skin)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b4 <- ggplot(PimaV, aes("var", bmi)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b5 <- ggplot(PimaV, aes("var", ped)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> b6 <- ggplot(PimaV, aes("var", age)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   geom_boxplot()
> grid.arrange(b1, b2, b3, b4, b5, b6, ncol = 6)
> # enforce y-range from -5 to 7.5
> PimaV <- select(Pima.tr2, glu:age)
> PimaV <- as.data.frame(scale(PimaV))
> b1 <- ggplot(PimaV, aes("var", glu)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   ylim(-5, 7.5) +
+   geom_boxplot()
> b2 <- ggplot(PimaV, aes("var", bp)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   ylim(-5, 7.5) +
+   geom_boxplot()
> b3 <- ggplot(PimaV, aes("var", skin)) +
```

```
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   ylim(-5, 7.5) +
+   geom_boxplot()
> b4 <- ggplot(PimaV, aes("var", bmi)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   ylim(-5, 7.5) +
+   geom_boxplot()
> b5 <- ggplot(PimaV, aes("var", ped)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   ylim(-5, 7.5) +
+   geom_boxplot()
> b6 <- ggplot(PimaV, aes("var", age)) +
+   xlab("") +
+   scale_x_discrete(breaks = NULL) +
+   ylim(-5, 7.5) +
+   geom_boxplot()
> grid.arrange(b1, b2, b3, b4, b5, b6, ncol = 6)
> ## ggplot2
>
> # alternatively: reshape the data first
> PimaV <- select(Pima.tr2, glu:age)
> PimaV <- as.data.frame(scale(PimaV))
> head(PimaV)
```

```
         glu         bp        skin        bmi        ped        age
1 -1.2576269 -0.3691372 -0.09873518 -0.2853660 -0.2432339 -0.7856735
2  2.3743082 -0.1982624  0.32925852 -1.0708354 -0.9255154  1.8917775
3 -1.5575115  0.8269863  1.01404844  0.5771102 -0.9492765  0.1643897
4  1.3746930  0.3143620  1.18524592  2.4406749 -0.5996496 -0.6129347
5 -0.5578963 -1.0526363 -0.35553140 -0.8706177 -1.0273485 -0.8720429
6 -0.8911014  0.3143620 -0.18433392  0.5463075 -0.1957118  1.6326693
```

```
> PimaV <- reshape(PimaV, idvar = "number",
+                  times = names(PimaV),
```

```
+                  timevar = "variable",
+                  varying = list(names(PimaV)),
+                  direction = "long")
> head(PimaV)

      variable        glu number
1.glu      glu -1.2576269      1
2.glu      glu  2.3743082      2
3.glu      glu -1.5575115      3
4.glu      glu  1.3746930      4
5.glu      glu -0.5578963      5
6.glu      glu -0.8911014      6

> ggplot(PimaV, aes(variable, glu)) +
+   geom_boxplot()
> # rearrange order of factor levels
> PimaV <- select(Pima.tr2, glu:age)
> PimaV <- as.data.frame(scale(PimaV))
> PimaV <- reshape(PimaV, idvar = "number",
+                  ids = row.names(PimaV),
+                  times = names(PimaV),
+                  timevar = "variable",
+                  varying = list(names(PimaV)),
+                  v.names = "data",
+                  direction = "long")
> PimaV$variable <- as.factor(PimaV$variable)
> PimaV <- within(PimaV, newvariable <-
+                 factor(variable, levels = c("glu", "bp", "skin", "bmi", "ped", "age"
> ggplot(PimaV, aes(newvariable, data)) +
+   geom_boxplot()
> # remove axis labels
> PimaV <- select(Pima.tr2, glu:age)
> PimaV <- as.data.frame(scale(PimaV))
> PimaV <- reshape(PimaV, idvar = "number",
+                  ids = row.names(PimaV),
+                  times = names(PimaV),
```

```
+                   timevar = "variable",
+                   varying = list(names(PimaV)),
+                   v.names = "data",
+                   direction = "long")
> PimaV$variable <- as.factor(PimaV$variable)
> PimaV <- within(PimaV, newvariable <-
+                  factor(variable, levels = c("glu", "bp", "skin", "bmi", "ped", "age"
> ggplot(PimaV, aes(newvariable, data)) +
+   xlab("") +
+   ylab("") +
+   geom_boxplot()
> # change symbol and color for outliers
> PimaV <- select(Pima.tr2, glu:age)
> PimaV <- as.data.frame(scale(PimaV))
> PimaV <- reshape(PimaV, idvar = "number",
+                   ids = row.names(PimaV),
+                   times = names(PimaV),
+                   timevar = "variable",
+                   varying = list(names(PimaV)),
+                   v.names = "data",
+                   direction = "long")
> PimaV$variable <- as.factor(PimaV$variable)
> PimaV <- within(PimaV, newvariable <-
+                  factor(variable, levels = c("glu", "bp", "skin", "bmi", "ped", "age"
> ggplot(PimaV, aes(newvariable, data)) +
+   xlab("") +
+   ylab("") +
+   geom_boxplot(outlier.shape = 16, outlier.color = "red")
> # interactive version
> ggplotly()
```

2.9 Figure 1.9

```
> ## ----pimaSpl---- Fig 1.9
>
> # basic graphic
```

```
> PimaV <- select(Pima.tr2, glu:age)
> ggpairs(PimaV)
> # no visible changes, so these are the defaults
> PimaV <- select(Pima.tr2, glu:age)
> ggpairs(PimaV, diag = list(continuous = "density"),
+         axisLabels = "show")
> # interactive version
> ggplotly()
> ## BaseR
>
> # basic scatterplot matrix
> pairs(PimaV)
```

2.10 Additional Par and Layout Options for baseR

```
> # Assume we want to plot 6 different plots at the same time
>
> myPlots <- function(n = 6) {
+   set.seed(7777)
+   sapply(1:n, function(x) hist(rnorm(100, sd = x),
+                                xlab = "x Values",
+                                xlim = c(-20, 20),
+                                main = paste("Histogram", x)))
+ }
> dev.off()

null device
          1

> myPlots()

         [,1]              [,2]              [,3]
breaks   Numeric,13        Numeric,7         Numeric,8
counts   Integer,12        Integer,6         Integer,7
density  Numeric,12        Numeric,6         Numeric,7
mids     Numeric,12        Numeric,6         Numeric,7
```

```
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> # change layout with par() function
>
> dev.off()

null device
          1

> par(mfrow = c(3, 2))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> dev.off()
```

```
null device
          1

> par(mfrow = c(2, 3))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> # adjust spacing around each plot
>
> dev.off()

null device
          1

> par(mfrow = c(2, 3), mar = c(0, 0, 0, 0))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
```

```
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
```

```
> dev.off()
```

```
null device
          1
```

```
> par(mfrow = c(2, 3), mar = c(2, 1, 0, 0))
> myPlots()
```

```
         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
```

```
> dev.off()
```

```
null device
          1
```

```
> par(mfrow = c(2, 3), mar = c(3, 2, 2, 0))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> dev.off()

null device
          1

> par(mfrow = c(2, 3), mar = c(4, 4, 2, 0))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
```

```
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> mtext("Histograms with 6 different SDs",
+       outer = TRUE,
+       side = 3)
> # allow space for a title above the 6 plots
>
> dev.off()

null device
          1

> par(mfrow = c(2, 3), mar = c(4, 4, 2, 0), oma = c(0, 0, 2, 0))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> mtext("Histograms with 6 different SDs",
+       outer = TRUE,
+       side = 3)
> dev.off()
```

```
null device
          1

> par(mfrow = c(2, 3), mar = c(4, 4, 2, 0), oma = c(0, 0, 2, 0))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> mtext("Histograms with 6 different SDs",
+       outer = TRUE,
+       side = 3,
+       cex = 1.5)
> dev.off()

null device
          1

> par(mfrow = c(2, 3), mar = c(4, 4, 2, 0), oma = c(0, 0, 3, 0))
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
```

```
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> mtext("Histograms with 6 different SDs",
+       outer = TRUE,
+       side = 3,
+       line = 1,
+       cex = 1.5)
> # general layouts with layout() function
>
> dev.off()

null device
          1

> graphics::layout(matrix(c(1, 1, 1, 2, 2, 3,
+                           4, 5, 5, 6, 6, 6),
+                         2, 6, byrow = TRUE))
> layout.show(6)
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
```

```
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
breaks   Numeric,12           Numeric,8            Numeric,8
counts   Integer,11           Integer,7            Integer,7
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE

> ### a rather complicated layout for 22 plots
>
> dev.off()

null device
          1

> n3 <- rep(0, 3)
> n4 <- rep(0, 4)
> graphics::layout(matrix(c(n4, rep(1, 5), n3, rep(2, 5), n3, rep(3, 5), n3, rep(4, 5)
+                            rep(5, 5), n3, rep(6, 5), n3, rep(7, 5), n3, rep(8, 5), n3
+                            n4, rep(10, 5), n3, rep(11, 5), n3, rep(12, 5), n3, rep(13
+                            rep(14, 5), n3, rep(15, 5), n3, rep(16, 5), n3, rep(17, 5)
+                            n4, rep(19, 5), n3, rep(20, 5), n3, rep(21, 5), n3, rep(22
+                          nrow = 5, byrow = TRUE))
> layout.show(22)
> par(mar = c(0, 0, 0, 0)) # should be adjusted to something more meaningful
> myPlots()

         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]                 [,5]                 [,6]
```

```
breaks   Numeric,12          Numeric,8           Numeric,8
counts   Integer,11          Integer,7           Integer,7
density  Numeric,11          Numeric,7           Numeric,7
mids     Numeric,11          Numeric,7           Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                TRUE                TRUE
```

```
> myPlots()
```

```
         [,1]                [,2]                [,3]
breaks   Numeric,13          Numeric,7           Numeric,8
counts   Integer,12          Integer,6           Integer,7
density  Numeric,12          Numeric,6           Numeric,7
mids     Numeric,12          Numeric,6           Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                TRUE                TRUE
         [,4]                [,5]                [,6]
breaks   Numeric,12          Numeric,8           Numeric,8
counts   Integer,11          Integer,7           Integer,7
density  Numeric,11          Numeric,7           Numeric,7
mids     Numeric,11          Numeric,7           Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                TRUE                TRUE
```

```
> myPlots()
```

```
         [,1]                [,2]                [,3]
breaks   Numeric,13          Numeric,7           Numeric,8
counts   Integer,12          Integer,6           Integer,7
density  Numeric,12          Numeric,6           Numeric,7
mids     Numeric,12          Numeric,6           Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                TRUE                TRUE
         [,4]                [,5]                [,6]
breaks   Numeric,12          Numeric,8           Numeric,8
counts   Integer,11          Integer,7           Integer,7
```

```
density  Numeric,11           Numeric,7            Numeric,7
mids     Numeric,11           Numeric,7            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
```

```
> myPlots(4)
```

```
         [,1]                 [,2]                 [,3]
breaks   Numeric,13           Numeric,7            Numeric,8
counts   Integer,12           Integer,6            Integer,7
density  Numeric,12           Numeric,6            Numeric,7
mids     Numeric,12           Numeric,6            Numeric,7
xname    "rnorm(100, sd = x)" "rnorm(100, sd = x)" "rnorm(100, sd = x)"
equidist TRUE                 TRUE                 TRUE
         [,4]
breaks   Numeric,12
counts   Integer,11
density  Numeric,11
mids     Numeric,11
xname    "rnorm(100, sd = x)"
equidist TRUE
```

```
> dev.off()
```

```
null device
          1
```

2.11 Exercise 1.1

Iris: How would you describe this histogram of sepal width?

```
> ## ----irisEx---- Ex. 1
> ggplot(iris, aes(Sepal.Width)) +
+   geom_histogram(binwidth = 0.1)
> summary(iris$Sepal.Width)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.800   3.000   3.057   3.300   4.400
```

2.12 Exercise 1.2

Pima Indians: Summarize what this barchart shows:

```
> ## ----PimaEx---- Ex. 2
> ggplot(Pima.tr2, aes(type)) +
+   geom_bar()
```

2.13 Exercise 1.3

Pima Indians: Why is the upper left of this plot of numbers of pregnancies against age empty?

```
> ## ----Pima2Ex---- Ex. 3
> ggplot(Pima.tr2, aes(age, npreg)) +
+   geom_point()
```

2.14 Exercise 1.4

Estimating the speed of light: There are 100 estimates of the speed of light made by Michelson in 1879, composed of 5 groups of 20 experiments each (dataset michelson in the MASS package).

(i) What plot would you draw for showing the distribution of all the values together? What conclusions would you draw?

(ii) What plots might be useful for comparing the estimates from the 5 different experiments? Do the results from the 5 experiments look similar?

```
Michelson's Speed of Light Data

Description
Measurements of the speed of light in air, made between 5th June and 2nd July, 1879.
The data consists of five experiments, each consisting of 20 consecutive runs.
The response is the speed of light in km/s, less 299000. The currently accepted
value, on this scale of measurement, is 734.5.

Usage
michelson
```

```
> data(michelson, package = "MASS")
> class(michelson)
```

```
[1] "data.frame"
```

```
> head(michelson)
```

```
  Speed Run Expt
1   850   1    1
2   740   2    1
3   900   3    1
4  1070   4    1
5   930   5    1
6   850   6    1
```

```
> summary(michelson)
```

```
     Speed                  Run       Expt
 Min.   : 620.0   1      : 5   1:20
 1st Qu.: 807.5   2      : 5   2:20
 Median : 850.0   3      : 5   3:20
 Mean   : 852.4   4      : 5   4:20
 3rd Qu.: 892.5   5      : 5   5:20
 Max.   :1070.0   6      : 5
                  (Other):70
```

```
> ggplot(michelson, aes(Speed)) +
+   geom_histogram()
> ggplot(michelson, aes(Speed)) +
+   geom_histogram(binwidth = 25)
> ggplot(michelson, aes(Speed)) +
+   geom_histogram(binwidth = 50)
> ggplot(michelson, aes(Speed)) +
+   geom_histogram(binwidth = 50) +
+   facet_wrap(~ Expt, ncol = 3)
> ggplot(michelson, aes(Expt, Speed)) +
+   geom_boxplot()
> summary(michelson)
```

```
    Speed              Run      Expt
 Min.   : 620.0   1      : 5   1:20
 1st Qu.: 807.5   2      : 5   2:20
 Median : 850.0   3      : 5   3:20
 Mean   : 852.4   4      : 5   4:20
 3rd Qu.: 892.5   5      : 5   5:20
 Max.   :1070.0   6      : 5
                  (Other):70
```

```
> summary(michelson[michelson$Expt != "1", ])
```

```
    Speed             Run      Expt
 Min.   :620.0   1      : 4   1: 0
 1st Qu.:800.0   2      : 4   2:20
 Median :840.0   3      : 4   3:20
 Mean   :838.2   4      : 4   4:20
 3rd Qu.:880.0   5      : 4   5:20
 Max.   :970.0   6      : 4
                 (Other):56
```

```
>
```

Scans//Cartoonstock_ArtistRetreat.jpg

Figure 11: http://www.cartoonstock.com/blowup_stock.asp?imageref=hsc3714&artist=Schwadron,+Harley&topic=statistics+, Cartoon.

— THE END —

Scans//Cartoonstock_FallingArrow.jpg

Figure 12: `http://www.cartoonstock.com/blowup_stock.asp?imageref=vsh0184&artist=Shirvanian,+Vahan&topic=statistics+`, Cartoon.