D

Answers to exercises

1.1 To insert 1.23 between `x[7]` and `x[8]`:

```
x <- round(rnorm(20,2))
z <- c(x[1:7],1.23,x[8:20])
```

or, more generally, to insert `v` just after index `k`

```
v <- 1.23; k <- 7
i <- seq(along=x)
z <- c(x[k<=i], v, x[k>i])
```

1.2 Here is one way:

```
x <- y <- c(7, 9, NA, NA, 13)
all(is.na(x) == is.na(y)) && all((x == y)[!is.na(x)])
```

Notice the use of && so same `NA` pattern is guaranteed in 2nd part.

1.3 Factor `x` gets treated as integer codes. Useful when selecting plot symbols, etc.

```
x <- factor(c("Huey", "Dewey", "Louie", "Huey"))
y <- c("red","green","blue")
x
y[x]
```

1.4

```
library(Rx)
data(thuesen)
attach(thuesen)
```

```
f <- cut(blood.glucose, c(4, 7, 9, 12, 20))
levels(f) <- c("low", "intermediate", "high", "very high")
```

1.5

```
data(juul)
juul.girl <- subset(juul, age >=7 & age < 14 & sex == 2)
summary(juul.girl)
```

("`age >=7 & age < 14 & sex == 2`" is an acceptable answer.)

1.6 The levels with the same name are collapsed into one.

1.7 `sapply(1:10, function(i) mean(rnorm(100)))`

1.8 (First part only) Use

```
write.table(thuesen, file="foo.txt")
```

and do as suggested, or (same effect)

```
write.table(thuesen, file="foo.txt", na=".")
read.table("foo.txt", na.strings=".")
```

2.1

```
1 - pnorm(3)
1 - pnorm(42, mean=35, sd=6)
dbinom(10, size=10, prob=0.8)
punif(0.9) # this one is obvious...
1 - pchisq(6.5, df=2)
```

You might use `lower.tail=FALSE` instead of subtracting from 1 in (a),(b), and (e).

2.2

```
qnorm(1-.05/2)
qnorm(1-.01/2)
qnorm(1-.005/2)
qnorm(1-.001/2)
qnorm(.25)
qnorm(.75)
```

Again, `lower.tail` can be used in the first four cases.

2.3 `dbinom(0, size=10, prob=.2)`

2.4 `ifelse(rbinom(10,1,.5) == 1, "H", "T")` or `c("H", "T")[1 + rbinom(10,1,.5)]`

3.1 (E.g.)

```
x <- 1:5 ; y <- rexp(5,1) ; par(mfrow=c(2,2))
plot(x, y, pch=15) # filled square
plot(x, y, type="b", lty="dotted")
```

```
plot(x, y, type="b", lwd=3)
plot(x, y, type="o", col="blue")
```

3.2 Use a filled symbol and set the fill:

```
plot(rnorm(10),type="o", pch=21, bg="white")
```

3.3 For a generic solution you need to peek inside the return value of `qqnorm`, but you could of course just eyeball the relevant limits.

```
x1 <- rnorm(20)
x2 <- rnorm(10)+1
q1 <- qqnorm(x1, plot.it=F)
q2 <- qqnorm(x2, plot.it=F)
xr <- range(q1$x, q2$x)
yr <- range(q1$y, q2$y)
qqnorm(x1, xlim=xr, ylim=yr)
points(q2, col="red")
```

(Notice how easily you can plot a list with an `x` and a `y` component. `qqnorm` is used for the basic plot to get the labels right.) Setting `type="l"` gives a messy plot because the values are not plotted in order. The remedy is to use `sort(x1)` and `sort(x2)`.

3.4 Breaks are at integer values, as are data. Data on the boundary are counted in with the column to the left, effectively shifting the histogram half a unit left. The `truehist` function allows you to specify a better set of breaks.

```
data(react)
hist(react)
library(MASS)
truehist(react,h=1,x0=.5)
```

3.5

```
z <- runif(5)
curve(quantile(z,x), from=0, to=1)
```

4.1 Yes (some discretization; two weak outliers, one at each end); Yes ($t = -7.75, p = 1.1 \times 10^{-13}$)

```
qqnorm(react)
t.test(react)
```

4.2 `t.test(vital.capacity~group, conf=0.99, data=vitcap)`; Notice that `age` also differs by group.

4.3

```
wilcox.test(react)
wilcox.test(vital.capacity~group, data=vitcap)
```

4.4

```
data(intake); attach(intake) ; par(mfrow=c(2,2))
plot(post ~ pre) ; abline(0,1)
plot((post+pre)/2, post - pre,
    ylim=range(0,post-pre)); abline(h=0)
hist(post-pre)
qqnorm(post-pre)
detach(intake)
```

4.5 Note: The outliers are the first and last observation in the (sorted) data vector, and we use a rather crude way of removing them below. You might want to think of something more elegant.

```
data(react)
shapiro.test(react)
shapiro.test(react[-c(1,334)])
qqnorm(react[-c(1,334)])
```

The test comes out highly significant even with outliers removed, because it picks up the discretization effect in the otherwise nearly straight-line `qqnorm` plot.

4.6 You'd expect that the two groups had similar differences, but with sign depending on which treatment was given first.

```
t.test(vas.active, vas.plac, paired=TRUE)
t.test((vas.active-vas.plac)[grp==1],
       (vas.plac-vas.active)[grp==2])
```

The first is a simple paired t test, the second test is corrected for period effect (which in this case is quite small).

4.7

```
t.test(rnorm(25))$p.value       #repeat 10x
t.test(rt(25,df=2))$p.value     #repeat 10x
t.test(rexp(25), mu=1)$p.value  #repeat 10x
x <- replicate(5000, t.test(rexp(25), mu=1)$p.value)
qqplot(sort(x),ppoints(5000),type="l",log="xy")
```

5.1

```
data(rmr)
fit <- lm(metabolic.rate~body.weight, data=rmr)
summary(fit)
811.2267 + 7.0595 * 70 # , or:
predict(fit, newdata=data.frame(body.weight=70))
qt(.975,42)
7.0595 + c(-1,1) * 2.018 * 0.9776 # , or:
confint(fit)
```

5.2 `summary(lm(sqrt(igf1)~age, data=juul, subset=age>25))`

5.3

```
data(malaria)
summary(lm(log(ab)~age, data=malaria))
plot(log(ab)~age, data=malaria)
```

5.4 (This could be elaborated by wrapping the random number generation in a function, etc.)

```
rho <- .90 ; n <- 100
x <- rnorm(n)
y <- rnorm(n, rho * x, sqrt(1 - rho^2))
plot(x, y)
cor.test(x, y)
cor.test(x, y, method="spearman")
cor.test(x, y, method="kendall")
```

6.1

```
data(zelazo)
walk <- unlist(zelazo) # or c(..,recursive=TRUE)
group <- factor(rep(1:4,c(6,6,6,5)), labels=names(zelazo))
summary(lm(walk ~ group))
t.test(zelazo$active,zelazo$ctr.8w) # first vs. last
t.test(zelazo$active,unlist(zelazo[-1])) # first vs. rest
```

6.2 A and C differs, B intermediate, not significantly different from either. The B-C comparison is not available from the summary, but due to the balanced design, the standard error of that difference is 0.16656 too.

```
data(lung)
fit <- lm(volume~method+subject, data=lung)
anova(fit)
summary(fit)
```

6.3

```
kruskal.test(walk ~ group)
wilcox.test(zelazo$active,zelazo$ctr.8w) # first vs. last
wilcox.test(zelazo$active,unlist(zelazo[-1])) # first vs. rest
friedman.test(volume ~ method | subject, data=lung)
wilcox.test(lung$volume[lung$method=="A"],
            lung$volume[lung$method=="C"], paired=TRUE) # etc.
```

6.4 (Only square root transform shown, do likewise for log transformed and untransformed data)

```
tapply(sqrt(igf1),tanner, sd, na.rm=TRUE)
plot(sqrt(igf1)~jitter(tanner))
oneway.test(sqrt(igf1)~tanner)
```

Transformations don't make much of a difference, but in all cases, strong age-effects are being ignored.

7.1 With 10 patients, $p = 0.1074$. Need 14 or more for significance at level 0.05.

```
binom.test(0,10, p=.20, alt="less")
binom.test(0,13, p=.20, alt="less")
binom.test(0,14, p=.20, alt="less")
```

7.2 `prop.test(c(210,122),c(747,661))`

7.3 Confidence interval is $(-0.08462185, 0.50657307)$

```
M <- matrix(c(23,7,18,13),2,2)
chisq.test(M)
fisher.test(M)
prop.test(M)
```

7.4 A simplified analysis, using `fisher.test` because of the small cell counts:

```
tbl <- c(42, 157, 47, 62, 4, 15, 4, 1, 8, 28, 9, 7)
dim(tbl) <- c(2,2,3)
dimnames(tbl) <- list(c("A","B"),
                      c("not pierced","pierced"),
                      c("ok","broken","cracked"))
ftable(tbl)
fisher.test(tbl["B",,]) # slice analysis
fisher.test(tbl["A",,])
fisher.test(margin.table(tbl,2:3)) # marginal
```

You may wish to check that there is little or no effect of egg size on breakage, so that the marginal analysis is defensible. You could also try collapsing the"broken" and "cracked" categories.

7.5 The curve shows substantial discontinuities where probability mass is shifted from one tail to the other, and also a number of local minima. A confidence region could be defined as those p that there is no significant evidence against at level α, but for some α that is not an interval.

```
p <- seq(0,1,0.001)
pval <- sapply(p,function(p)binom.test(3,15,p=p)$p.value)
plot(p,pval,type="l")
```

8.1 Estimated sample size: 6.29 or 8.06 per group depending on one- or two-sided testing. Approximate formula: 6.98 for the two-sided case. A small drop in power results from the unbalanced sampling.

```
power.t.test(power=.8,delta=.30,sd=.20)
power.t.test(power=.8,delta=.30,sd=.20,alt="one.sided")
(qnorm(.975)+qnorm(.8))^2*2*(.2/.3)^2 # approx. formula
power.t.test(n=8, delta=.30, sd=.20)   # power with eq.size
d2 <- .30 * sqrt(2/8) / sqrt(1/6+1/10) # corr.f.uneq. size
power.t.test(n=8, delta=d2, sd=.20)
```

8.2

```
power.prop.test(power=.9, p1=.6, p2=.75)
power.prop.test(power=.8, p1=.6, p2=.75)
```

8.3 (`dt` does actually allow `ncp` now!). Notice that the noncentral t distribution is asymmetric, with a rather heavy right tail.

```
mydt <- function(x,df,ncp)
   (pt(x+.001, df, ncp) - pt(x-.001, df, ncp)) * 500
curve(dt(x-3, 25), from=0, to=5)
curve(mydt(x, 25, 3), add=TRUE)
```

8.4 (Recent versions of R have `strict` argument.) This causes the power at zero effect size to be *half* the significance level, in contradition ot theory. For any relevant effect size, the difference is immaterial.

8.5 Approximately 0.50; exactly so if the variance is assumed known. The estimated SE in t tests complicates things.

9.1 The model with both diameters has a residual error of 0.107, compared to 0.128 using abdominal diameter alone and 0.281 with no predictors at all. If a fetus is scaled isotropically, a cubic relation with weight is expected, and you could speculate that this is reflected in the sum of coefficients when using log scales.

```
summary(lm(log(bwt) ~ log(bpd) + log(ad), data=secher))
summary(lm(log(bwt) ~ log(ad), data=secher))
```

9.2 If you use `data(tlc);attach(tlc)` you will have the `tlc` data frame in the global environment, where it will mask the variable inside `tlc`. Don't use the `attach` mechanism, or rename or remove `tlc` in the global environment.

```
pairs(tlc)
summary(lm(log(tlc) ~ ., data=tlc))
plot(lm(log(tlc) ~ ., data=tlc))
drop1(lm(log(tlc) ~ ., data=tlc))
drop1(lm(log(tlc) ~ . - age, data=tlc))
plot(log(tlc) ~ height,data=tlc)
plot(lm(tlc ~ .,data=tlc)) # slightly worse
```

9.3 The regression coefficient does not describe a slope, but rather a value to be added for females.

9.4 `age` is highly significant in the first analysis, but only borderline significant ($p = 0.06$) in the second analysis after removing `height` and `weight`. The two tests relate to the same model, but the number of observations differs.

```
summary(lm(sqrt(igf1) ~ age, data=juul2, subset=(age >= 25)))
anova(lm(sqrt(igf1) ~ age + weight + height,
```

```
        data=juul2, subset=(age >= 25)))
```

9.5 `sex` is treated as a binary indicator for girls. Notice that there are effects both of the mother's and of the child's size. The reason why height rather than weight of the mother enters into the equation is somewhat obscure, but one could speculate that weight is an unreliable indicator shortly after pregnancy.

```
summary(lm(dl.milk ~ . - no, data=kfm))
summary(lm(dl.milk ~ . - no - mat.weight, data=kfm))
summary(lm(dl.milk ~ . - no - mat.weight - sex, data=kfm))
summary(lm(dl.milk ~ weight + mat.height, data=kfm))
```

10.1 The hard bit is to set up the data in "long" format, with one `vas` observation per record.

```
ashina$subject <- factor(1:16)
attach(ashina)
act <- data.frame(vas=vas.active, subject, treat=1, period=grp)
plac <-data.frame(vas=vas.plac, subject, treat=0,
                  period=ifelse(grp==1,2,1))
ashina.long <- rbind(act, plac)
ashina.long$treat <- factor(ashina.long$treat)
ashina.long$period <- factor(ashina.long$period)

fit.ashina <- lm(vas ~ subject + period + treat, data=ashina.long)
drop1(fit.ashina)
anova(fit.ashina)

dd <- vas.active - vas.plac
t.test(dd[grp==1], -dd[grp==2], var.eq=T)
t.test(dd[grp==1], dd[grp==2], var.eq=T)
```

Notice that the unbalance in group sizes makes the tests for period and treatment effects order-dependent, and the ANOVA table somewhat misleading.

10.2

```
attach(tb.dilute)
anova(lm(reaction ~ animal + logdose))
ld <- c(0.5, 0, -0.5)[logdose]
anova(lm(reaction ~ animal + ld))
summary(lm(reaction ~ animal + ld))
4.7917 + 0.6039 * qt(c(.025,.975), 11)
# or:
confint(lm(reaction ~ animal + ld))["ld",]

slopes <- reaction[logdose==0.5] - reaction[logdose==-0.5]
t.test(slopes)

anova(lm(reaction ~ animal*ld))
```

Re. slope calculations: The formula $\hat{beta} = \sum xy / \sum x^2$ ($\bar{x} = 0$, obviously) reduces to taking differences. We also rely on data order.

The confidence interval is wider in the *t* test, reflecting that slopes may vary between rats and that there are fewer degrees of freedom for estimating the variation.
The final ANOVA contains a test for parallel slopes and the *F* statistic is less than one, so in these data, the slopes vary *less* than expected, and the DF must be the important issue for the confidence interval.

10.3 This can be varied indefinitely, but consider these examples

```
model.matrix(~ a:b)         ; lm(z ~ a:b)
model.matrix(~ a * b)       ; lm(z ~ a * b)
model.matrix(~ a:x)         ; lm(z ~ a:x)
model.matrix(~ a * x)       ; lm(z ~ a * x)
model.matrix(~ b * (x + y)) ; lm(z ~ b * (x + y))
```

R will reduce the set of design variables for an interaction term between categorical variables if a main effect is present. but not detect the singularity caused by the presence of the intercept. There's no singularities in either of the two cases involving a categorical and a continuous variable, but the first one has one parameter less (common-intercept model).
The last example has a "coincidental" singularity (`x` and `y` are proportional within each level of `b`) which R has no chance of detecting. You can easily see that the model matrix is singular since the sum of the last two columns (`b2:x` and `b2:y`) is proportional to the second (`b2`), and also the difference between the `x` and `y` columns is a linear combination of `b2` and `b2:x`.

10.4 The models can be illustrated by plotting the fitted values against time with separate symbols for each person, e.g.

```
tt <- c(20,30,60,90,0)[time]
plot(fitted(model4)~tt,pch=as.character(person))
```

With `model1` there is no imposed structure, `model2` is completely additive so that the individual traces are parallel to each other, `model3` allows the jump from the "pre" value to the value at 20 minutes to vary between individuals, and finally `model4` is like `model3` except that there is no change after 30 minutes (traces become horisontal). So `model3` is nested in `model1` and both `model2` and `model4` are nested in `model3`, but there is no nesting relation between `model2` and `model4`.

10.5
```
bp.obese <- transform(bp.obese,sex=factor(sex, labels=c("M","F")))
plot(log(bp) ~ log(obese), pch=c(20,21)[sex], data=bp.obese)
summary(lm(log(bp) ~ sex, data=bp.obese))
summary(lm(log(bp) ~ sex + log(obese), data=bp.obese))
summary(lm(log(bp) ~ sex*log(obese), data=bp.obese))
```

10.6
```
vitcap2 <- transform(vitcap2,group=factor(group,
                                          labels=c("exp>10",
                                          "exp<10", "unexp")))
attach(vitcap2)
```

```
plot(vital.capacity~age, pch=(20:22)[group])
vit.fit <- lm(vital.capacity ~ age*group)
summary(vit.fit)
drop1(vit.fit, test="F")
for (i in 1:3) abline(lm(vital.capacity ~ age,
                         subset=as.numeric(group)==i), lty=i)
legend(locator(1),legend=levels(group), pch=20:22, lty=1:3)
```

Notice that there is a significant interaction, i.e. the lines are *not* parallel.

10.7

```
juul.prepub <- subset(juul, tanner==1)
summary(lm(sqrt(igf1)~age, data=juul.prepub, subset= sex==1)) # boys
summary(lm(sqrt(igf1)~age, data=juul.prepub, subset= sex==2)) # girls
summary(lm(sqrt(igf1)~age*factor(sex), data=juul.prepub))
summary(lm(sqrt(igf1)~age+factor(sex), data=juul.prepub))
```

10.8

```
summary(fit.aicopt <- step(lm(dl.milk ~ . - no, data=kfm)))
plot(fit.aicopt)
kfm[32,]
summary(kfm)
summary(update(fit.aicopt,~ -sex))
plot(update(fit.aicopt, ~ . - sex - ml.suppl))
```

Observation 32 contains an extremely large value of `ml.suppl` and therefore has a large influences its regression coefficient. Without `ml.suppl` in the model, the Cook's distances are much smaller.

10.9

```
juulyoung <- subset(juul, age < 25)
juulyoung <- transform(juulyoung,
                sex=factor(sex), tanner=factor(tanner))
fit.untf <- lm(igf1 ~ age * sex * tanner, data=juulyoung,
               na.action=na.exclude)
plot(fitted(fit.untf) ~ age, data=juulyoung, col=c("red","green")[sex])
fit.log <- update(fit.untf, log(igf1) ~ .)
fit.sqrt <- update(fit.untf, sqrt(igf1) ~ .)
par(mfrow=c(2,2))
plot(fit.untf)
plot(fit.log)
plot(fit.sqrt)
```

11.1

```
summary(glm(mal~age+log(ab), binomial, data=malaria))
```

(you may also want to check for interaction)

11.2

```
attach(graft.vs.host)
type <- factor(type,labels=c("AML", "ALL", "CML"))
m1 <- glm(gvhd~rcpage+donage+type+preg+log(index), binomial)
m1a <- glm(gvhd~rcpage+donage+type+preg+index, binomial)
summary(m1)
summary(m1a)
```

It is seen that `log(index)` is more significant, but the model with `index` has a slightly better deviance. There is little hard evidence for either. The log transform has the advantage that it reduces the influence of two very large values of `index`.

```
drop1(m1, test="Chisq")
drop1(update(m1, ~ . - rcpage), test="Chisq")
drop1(update(m1, ~ . - rcpage - type), test="Chisq")
drop1(update(m1, ~ . - rcpage - type - preg), test="Chisq")
summary(m2 <- glm(gvhd~donage + log(index), binomial))
```

Notice that except for `log(index)` it is essentially arbitrary which variables end up in the final model. Altman (1991) treats the `type` classification as separate binary variables and gets a final model where ALL and AML are combined in to one group and includes `preg` but not `donage`.

11.3 (e.g.)

```
library(MASS) # unnecessary as of R 2.1.0
confint(m2)
## normal approximation:
est <-  coefficients(summary(m2))[,1]
se <-  coefficients(summary(m2))[,2]
est + cbind(qnorm(.025)*se, qnorm(.975)*se)
```

Notice that the `confint`-generated intervals lie assymmetrically around the estimate. In this case, both ends of the interval are shifted away from zero, in accordance with the fact that the deviance-based tests from `drop1` have lower p-values than the approximate t tests in `summary`.

11.4

```
counts <- c(13,40,157,40,21,61)
total <- c(108,264,375,310,181,162)
age <- gl(3,1,6)
type <- gl(2,3,6)
anova(glm(counts/total~age+type,weights=total, binomial),
      test="Chisq")
```

11.5

```
juul.girl <- subset(juul,age>8 & age<20 &
                    complete.cases(menarche))
logit.menarche <- glm(menarche~age+I(age^2)+I(age^3),
                      binomial, data=juul.girl)
probit.menarche <- glm(menarche~age+I(age^2)+I(age^3),
                       binomial(probit), data=juul.girl)
summary(logit.menarche)
summary(probit.menarche)
Age=seq(8,20,.1)
newages <- data.frame(age=Age)
p.logit <- predict(logit.menarche,newdata=newages,type="resp")
```

```
p.probit <- predict(probit.menarche,newdata=newages,type="resp")
matplot(Age,cbind(p.probit,p.logit),type="l")
```

12.1

```
attach(graft.vs.host)
plot(survfit(Surv(time,dead)~gvhd))
survdiff(Surv(time,dead)~gvhd)
summary(coxph(Surv(time,dead) ~ gvhd)) # for comparison
summary(coxph(Surv(time,dead) ~
             gvhd+log(index)+donage+rcpage+preg))
```

Subsequent elimination shows that `preg` might be a better predictor than `gvhd`.

12.2

```
cox1 <- coxph(Surv(days, status==1) ~
             log(thick) + sex + strata(ulc))
new <- data.frame(sex=2, thick=c(0.1, 0.2, 0.5))
svfit <-  surv fit(cox1,newdata=new)
plot(svfit[2], ylim=c(.985, 1))
```