

Introduction to R and RStudio

Part 3: Introduction to Descriptive and
Inferential Statistics with R

Rob Cribbie
Department of Psychology
York University

Example Data Set

- ▶ A researcher is interested in evaluating two therapies for perfectionism; specifically, investigating whether they will be effective in reducing levels of perfectionism
 - Levels of perfectionism are recorded at baseline (perf1), after 1 month of therapy (perf2) and after 2 months of therapy (perf3) for each experimental group (General Stress, CBT) and a control group
- ▶ At baseline the researcher also records levels of depression and the sex of each subject

Dataset

```
> head(dat)
```

	sex	group	dep1	perf1	perf2	perf3
1	m	cbt	40	78	72	60
2	m	cbt	56	66	55	51
3	m	cbt	108	82	80	80
4	m	cbt	98	75	70	72
5	m	cbt	67	71	63	54
6	m	cbt	70	73	62	54

|

Frequencies

- ▶ The 'table' function is helpful for obtaining frequencies for single or multiple variables

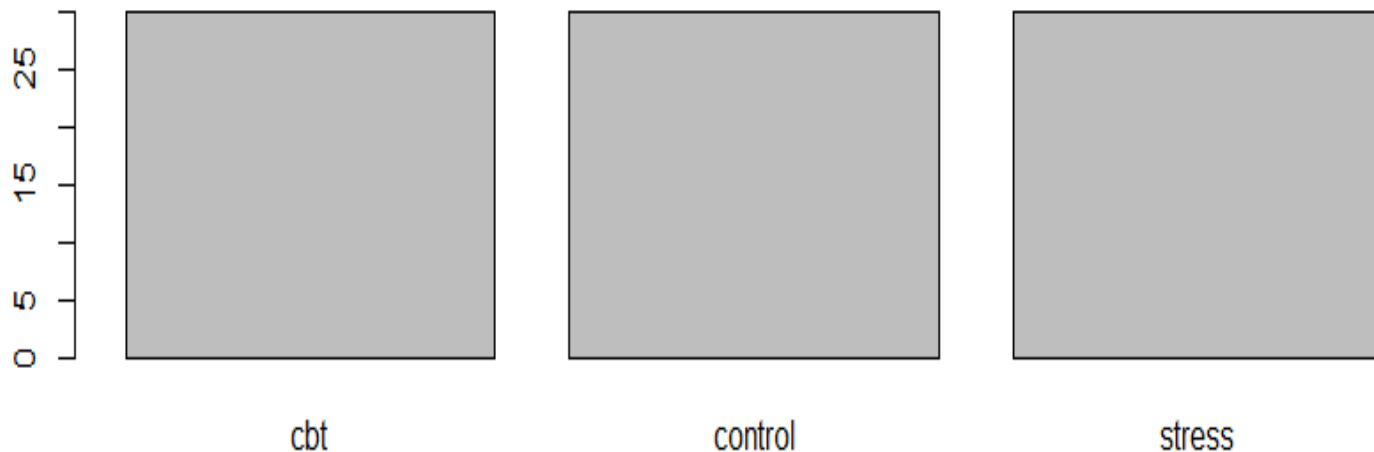
```
> table(dat$sex)           > table(dat$sex, dat$group)
```

```
f  m
40 50
|
```

	cbt	control	stress
f	13	15	12
m	17	15	18

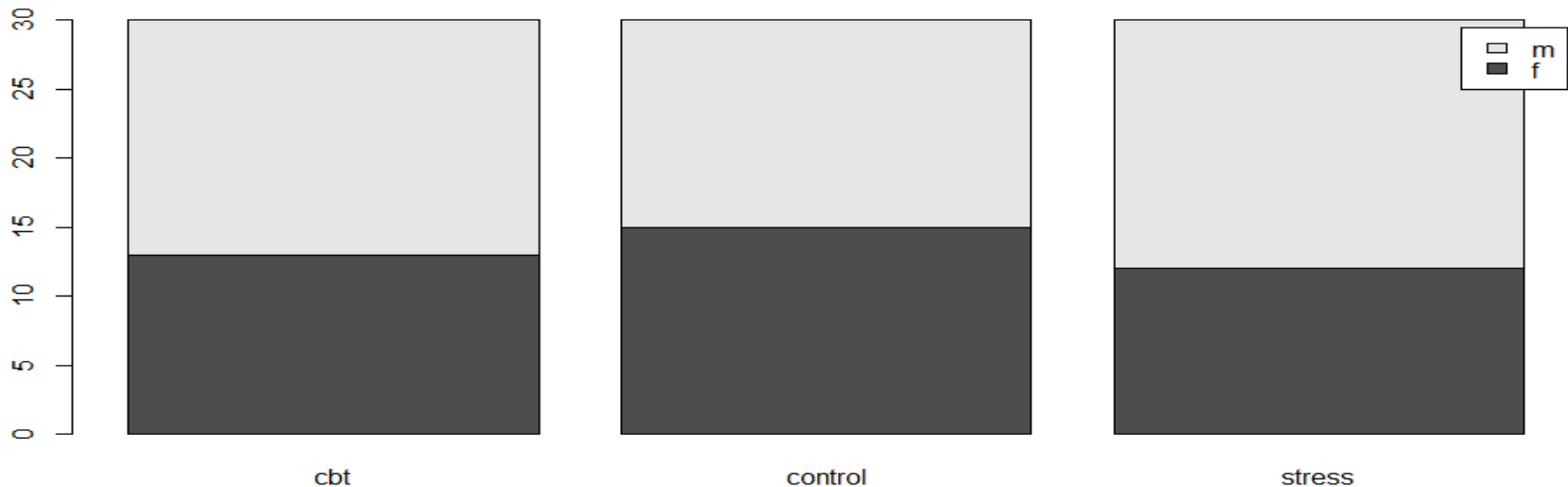
Frequencies

- ▶ We can also obtain simple graphical descriptions of frequencies
 - > `fgroup <- table(dat$group)`
 - > `barplot(fgroup)`



Frequencies

- ▶ Or, slightly more sophisticated graphical descriptions of frequencies
 - > `freq <- table(datsex, datgroup)`
 - > `barplot(freq, legend=rownames(freq))`



Basic Descriptive Statistics

- ▶ There are numerous ways to obtain basic descriptive statistics, for example:
 - `summary(x)`
 - `max(x)`
 - `min(x)`
 - `range(x)`
 - `mean(x)`
 - `median(x)`
 - `var(x)`
 - `sd(x)`
 - `quantile(x,probs=)`
- ▶ These can also be used with all the subsetting/indexing we previously discussed

Basic Descriptive Statistics

- ▶ However, there are also packages with functions for obtaining a lot of descriptive info quickly
 - E.g., 'describe' function in the 'psych' package

```
> library(psych)
> describe(dat)
```

```
vars  n  mean  sd  median  trimmed  mad  min  max  range
sex*   1 90  1.56  0.50   2.00    1.57   0.00  1.00  2.00  1.00
group* 2 90  2.00  0.82   2.00    2.00   1.48  1.00  3.00  2.00
dep1   3 90 81.51 16.79  80.08   80.83  17.61 49.25 129.48 80.24
perf1  4 90 80.53  9.79  80.24   80.09   8.73 59.67 111.19 51.52
perf2  5 90 72.55 13.39  71.96   72.06  11.65 43.80 109.34 65.54
perf3  6 90 69.50 14.83  68.09   68.54  15.17 40.81 119.09 78.28

      skew  kurtosis  se
sex*  -0.22    -1.97  0.05
group* 0.00    -1.53  0.09
dep1   0.43    -0.30  1.77
perf1  0.50     0.40  1.03
perf2  0.37     0.26  1.41
perf3  0.63     0.37  1.56
```


Basic Descriptive Statistics

- ▶ A similar function in the 'psych' package also allows us to obtain descriptive statistics separated by a grouping variable

```
> describeBy(dat, sex)
```

```
group: f
```

	vars	n	mean	sd	median	trimmed	mad
sex*	1	40	1.00	0.00	1.00	1.00	0.00
group*	2	40	1.98	0.80	2.00	1.97	1.48
dep1	3	40	81.36	18.32	80.62	80.61	20.18
perf1	4	40	79.47	10.28	79.33	79.24	8.28
perf2	5	40	72.30	13.70	72.96	71.76	13.72
perf3	6	40	69.66	14.76	68.21	68.78	15.22

This is just the first few lines, several statistics and the info for males is left out

Pearson Correlation

- ▶ Hypothesis #1: Are baseline depression and perfectionism scores correlated?

```
> cor.test(dat$perf1, dat$dep1)
```

```
Pearson's product-moment  
correlation
```

```
data: dat$perf1 and dat$dep1  
t = 6.047, df = 88, p-value =  
3.493e-08  
alternative hypothesis: true correlation  
is not equal to 0  
95 percent confidence interval:  
0.3770138 0.6733433  
sample estimates:  
cor  
0.541803
```

Rank-based Correlation

- ▶ Hypothesis #1: Are baseline depression and perfectionism scores correlated

- Spearman's rank-based correlation coefficient

```
> cor.test(dat$perf1, dat$dep1, method="spearman")
```

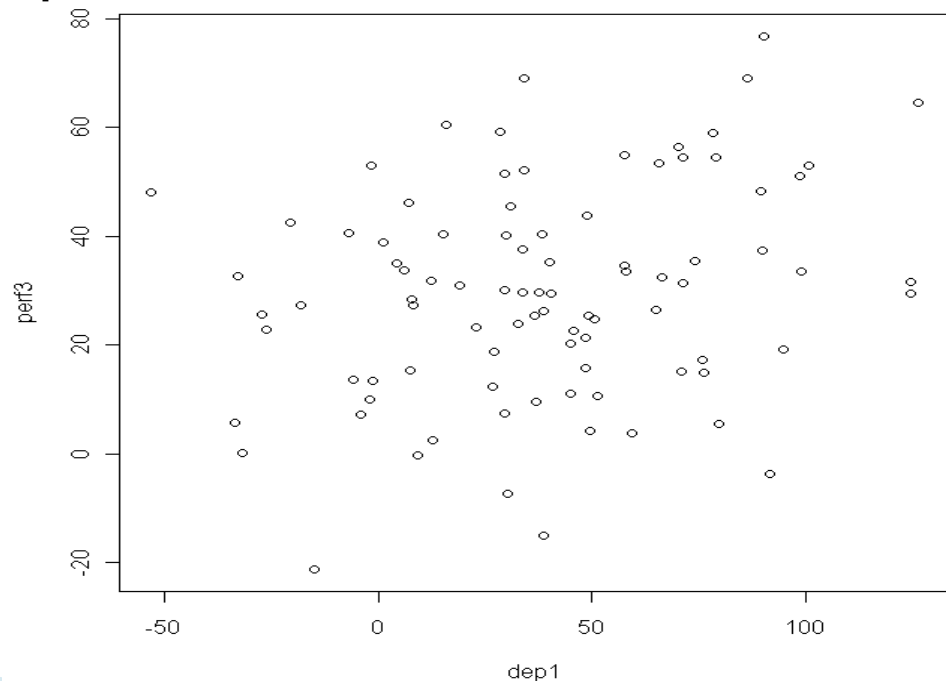
```
Spearman's rank correlation rho
```

```
data: dat$perf1 and dat$dep1  
S = 53726, p-value = 1.996e-08  
alternative hypothesis: true rho is not  
equal to 0  
sample estimates:  
rho  
0.5577561
```

Simple Regression

- ▶ Hypothesis #2: Can we predict posttest perfectionism scores from pretest depression scores?
- ▶ Scatterplot
 - `> plot(dat$dep1, dat$perf3)`

What happens with the 'plot' function depends on the nature of the variables



Simple Regression, cont'd

- ▶ Create a linear model object and print a summary of the results

```
> mod1<-lm(perf3 ~ dep1, data=dat)
> summary(mod1)
```

```
Call:
```

```
lm(formula = perf3 ~ dep1, data = dat)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-26.540	-10.605	-0.683	9.031	47.663

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	45.47798	7.38149	6.161	2.12e-08	***
dep1	0.29476	0.08871	3.323	0.0013	**

```
---
```

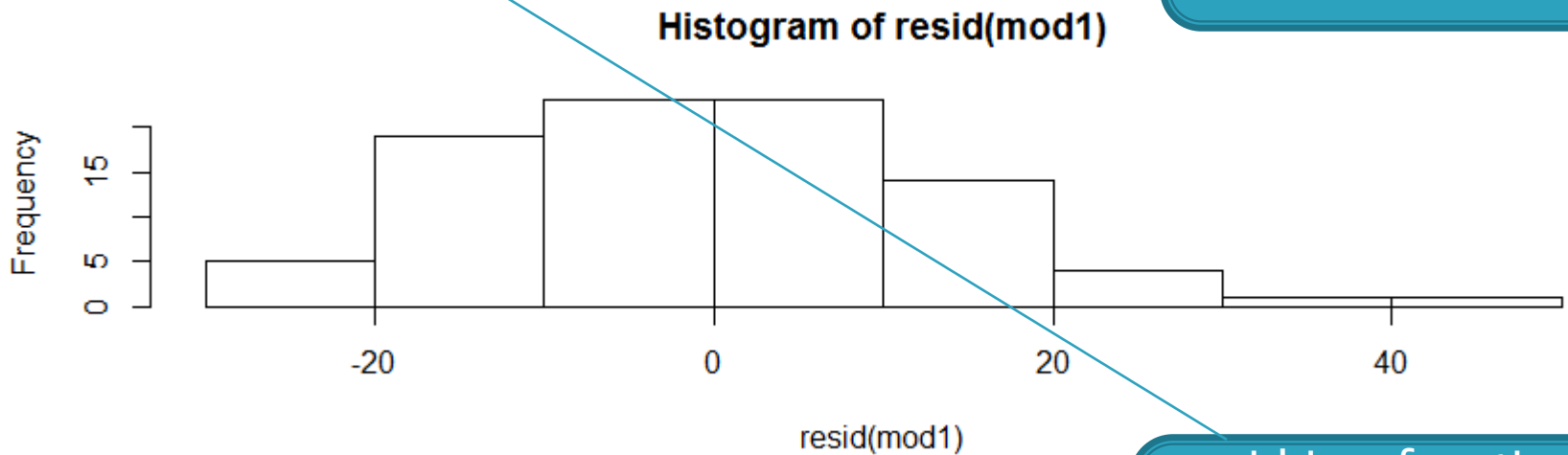
```
Signif. codes:
```

```
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 14.06 on 88 degrees of freedom
Multiple R-squared: 0.1115, Adjusted R-squared: 0.1014
F-statistic: 11.04 on 1 and 88 DF, p-value: 0.001301
```

Simple Regression, cont'd

- ▶ Diagnostics are available for identifying normality issues
 - `hist(resid(mod1))`



Normal probability (qq) plots are also available

`resid` is a function for extracting the residuals from a model object

Regression Diagnostics

- ▶ It is also easy to produce several diagnostic plots simply by typing:
 - `plot(mod)`
 - Where you replace 'mod' with the name of your model
- ▶ These plots include:
 - Residuals vs Fitted
 - Normal Probability Plot of the Residuals
 - Scale vs Location Plot
 - Residuals vs Leverage

Simple Regression, cont'd

▶ Regression Diagnostics

- We can compute Studentized Deleted Residuals to identify outlying cases in the solution
 - There are also tools for identifying cases with extreme leverage
- Below case 31 seems discrepant

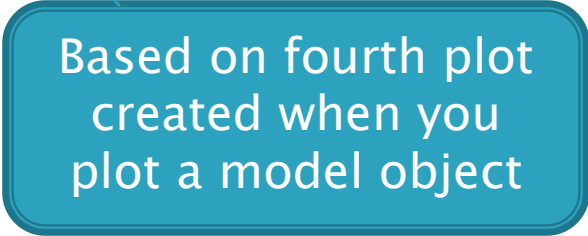
```
> library(car)
```

```
> outlierTest(mod1)
```

	rstudent	unadjusted	p-value	Bonferonni	p
31	3.643018		0.00045739		0.041165

Simple Regression, cont'd

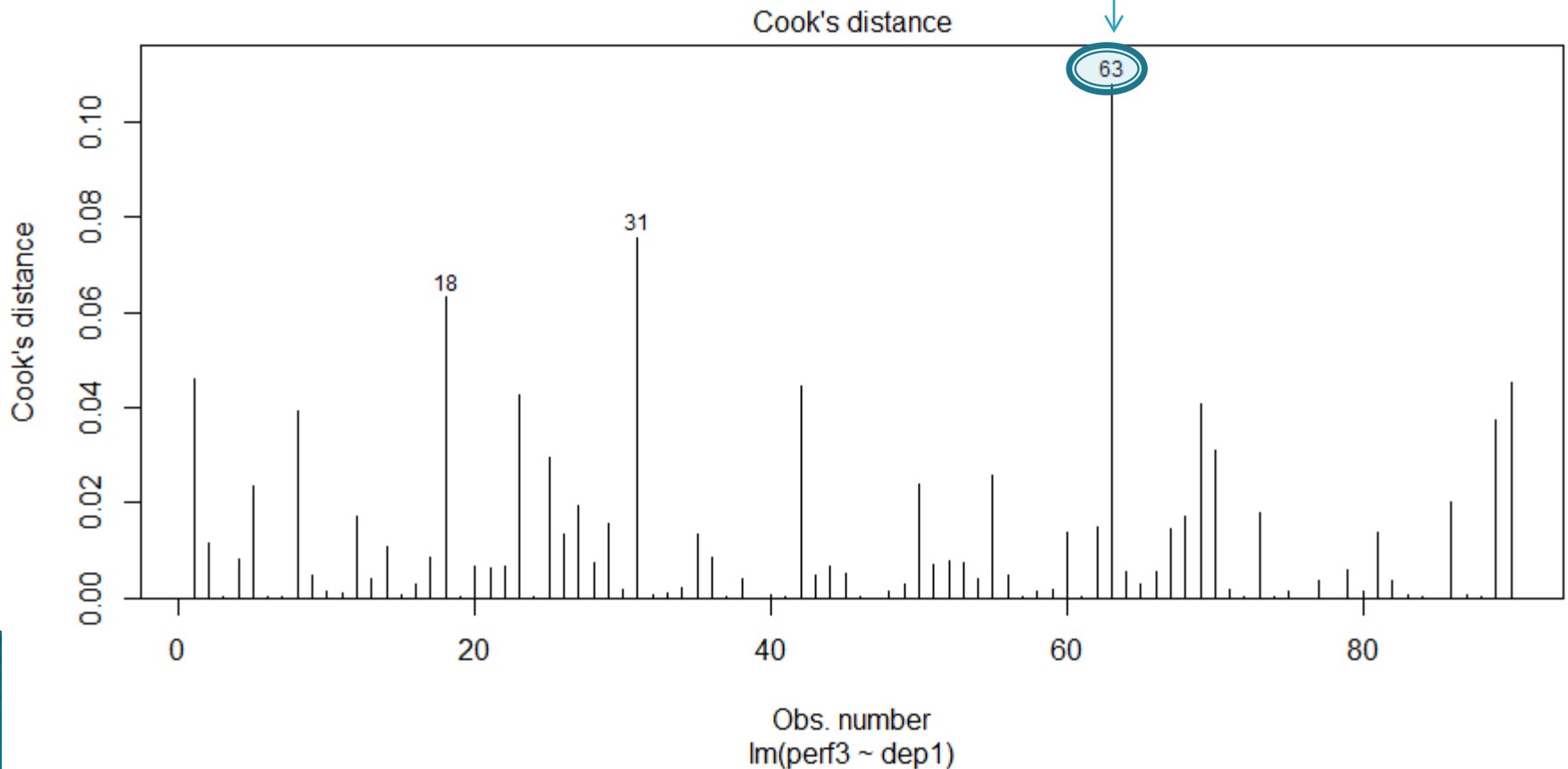
- ▶ It is also important to identify influential observations (e.g., Cook's Distance)
- ▶ This plot indicates which cases exceed a cutoff for Cook's distance of $4/(N - p)$ where p is the number of predictors
 - `>cutoff <- 4/(length(perf1)-1)`
 - `>plot(mod1, which=4, cook.levels=cutoff)`



Based on fourth plot
created when you
plot a model object

Identifying Influential Observations

Has a large influence on the model coefficients



Multiple Regression

- ▶ Hypothesis #3: Can posttest perfectionism scores be predicted from depression scores, controlling for pretest perfectionism?

```
> mod2<-lm(perf3 ~ dep1 + perf1, data=dat)
> summary(mod2)
```

Call:

```
lm(formula = perf3 ~ dep1 + perf1, data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-27.4451	-9.1199	-0.8234	10.2411	25.3162

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-4.75514	10.47354	-0.454	0.651
dep1	0.00531	0.08938	0.059	0.953
perf1	0.91674	0.15337	5.977	4.87e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

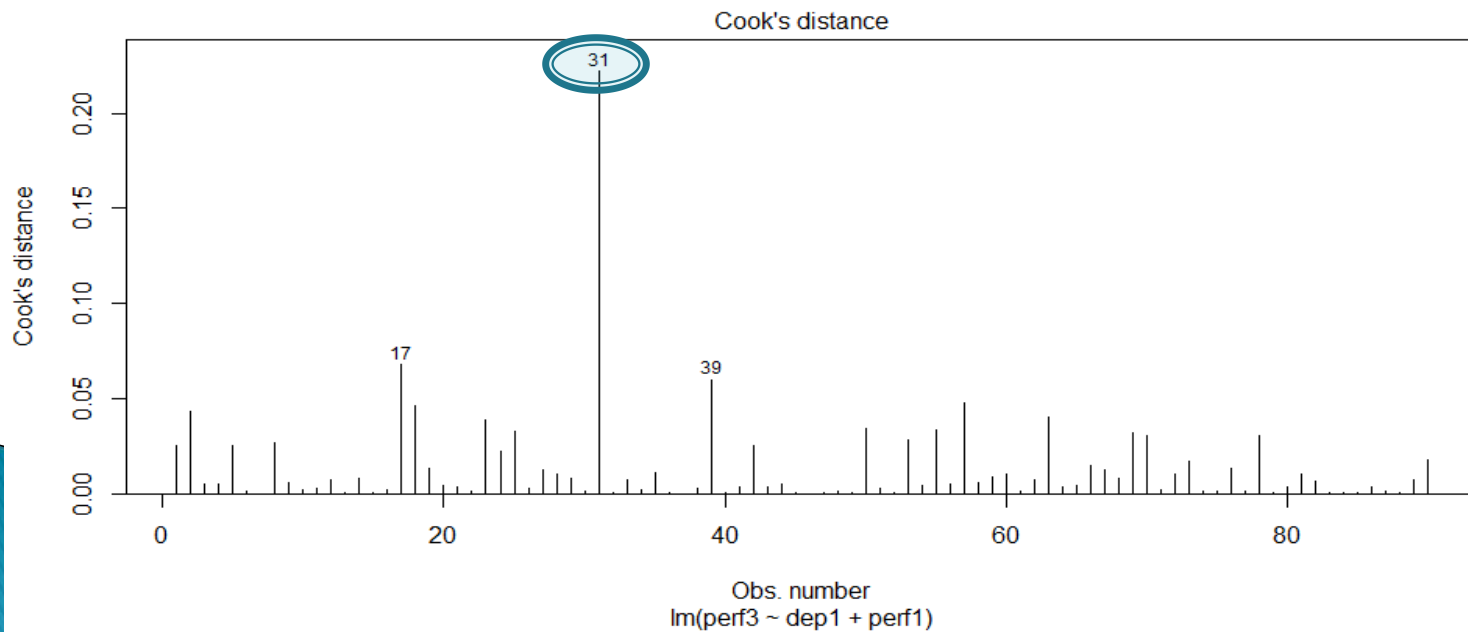
Residual standard error: 11.9 on 87 degrees of freedom

Multiple R-squared: 0.3701, Adjusted R-squared: 0.3557

F-statistic: 25.56 on 2 and 87 DF, p-value: 1.851e-09

Multiple Regression Diagnostics

- ▶ All of the same diagnostics used in a simple regression can also be applied in a multiple regression
- ▶ For example:
`plot(mod2,which=4,cook.levels=cutoff)`



Multiple Regression Diagnostics

- ▶ There are also diagnostics that are unique to multiple regression, such as collinearity diagnostics
 - Time 1 Depression and Perfectionism do not seem to be too highly related

```
> vif(mod2)
      dep1      perf1
1.415529  1.415529
. |
```

Interactions in Multiple Regression

```
> dat$dep1c<-dat$dep1-mean(dat$dep1)
> dat$perf1c<-dat$perf1-mean(dat$perf1)
> mod3<-lm(perf3 ~ dep1c + perf1c + dep1c|perf1c, data=dat)
> dat$dep1c<-dat$dep1-mean(dat$dep1)
> dat$perf1c<-dat$perf1-mean(dat$perf1)
> mod3<-lm(perf3 ~ dep1c + perf1c + dep1c:perf1c, data=dat)
> mod3<-lm(perf3 ~ dep1c*perf1c, data=dat)
> summary(mod3)
```

These two models are identical

We center the variables first to enhance interpretation

call:

```
lm(formula = perf3 ~ dep1c * perf1c, data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-27.1428	-8.2889	-0.1236	6.9596	23.4833

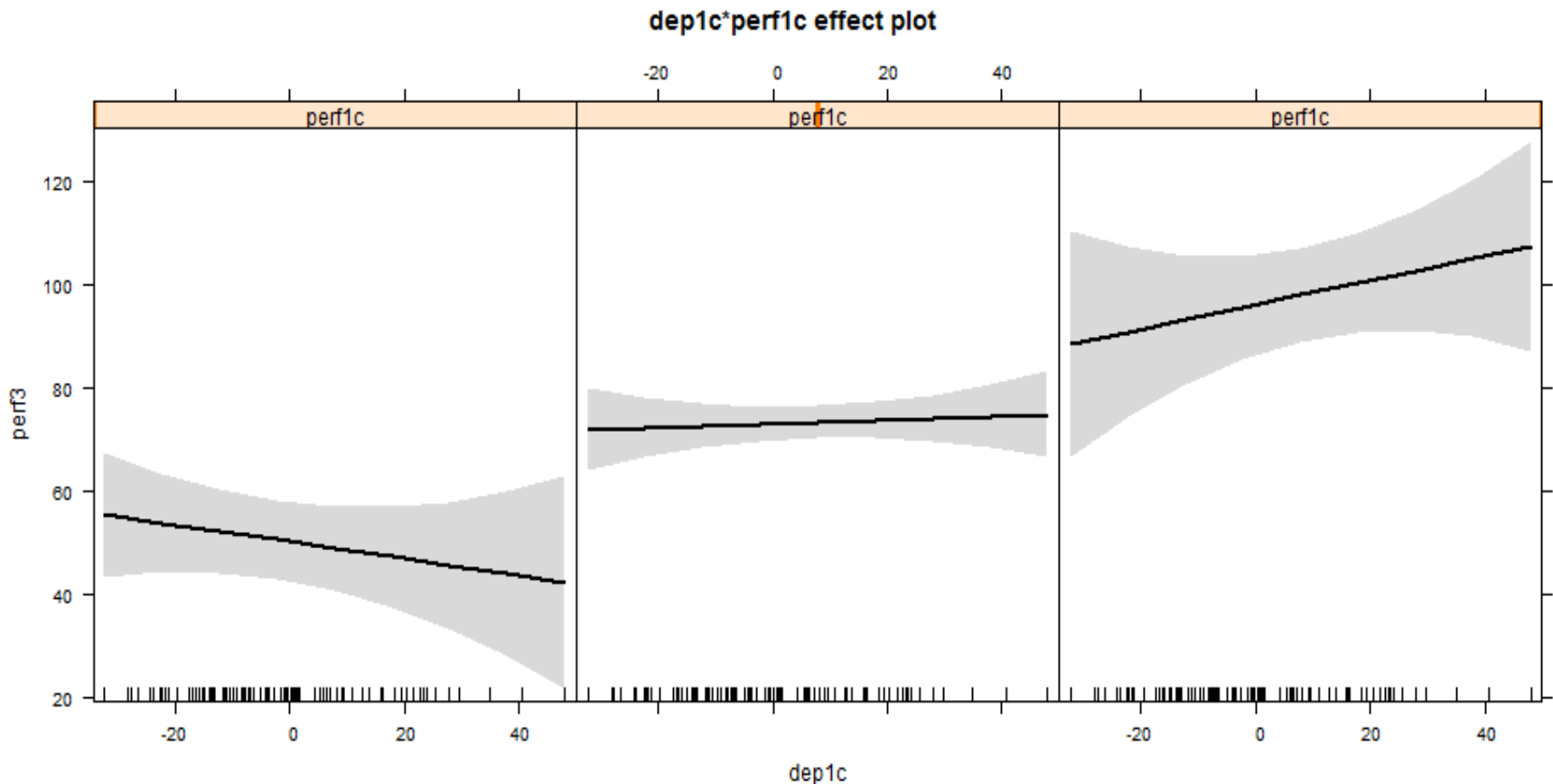
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	69.746849	1.296595	53.792	< 2e-16	***
dep1c	-0.050374	0.067761	-0.743	0.459	
perf1c	0.997633	0.143303	6.962	6.31e-10	***
dep1c:perf1c	-0.007919	0.005788	-1.368	0.175	

Interaction Plot

There are infinite numbers of cool stats, plots, etc. that you can find just by playing around with R

- `> library(effects)`
- `> plot(effect(term="dep1c:perf1c",mod=mod3,default.levels=3))`



Logistic Regression

- ▶ Imagine that we wanted to determine if a perfectionism diagnosis (normal, clinical) at time 1 is related to depression at time 1
- ▶ In this case we might want to use a logistic regression, a type of generalized linear model (glm)
- ▶ First, we will split our time 1 perfectionism scores into normal/clinical
 - `> library(Hmisc)`
 - `> dat$perfdich <- cut2(dat$perf1, cuts=90)`
 - `> levels(dat$perfdich) <- c("normal", "clinical")`

> 90 is the clinical cutoff

cut2 performs a split at the specified cutoff

Logistic Regression

```
> mod4<- glm(perfdich~dep1, family=binomial(link="logit"), data=dat)
> summary(mod4)
```

Call:

```
glm(formula = perfdich ~ dep1, family = binomial(link = "logit"),
     data = dat)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.4817	-0.5411	-0.3710	-0.2329	2.2472

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-7.70958	1.95033	-3.953	7.72e-05	***
dep1	0.06880	0.02088	3.295	0.000983	***

Generalized Linear
Model (glm)

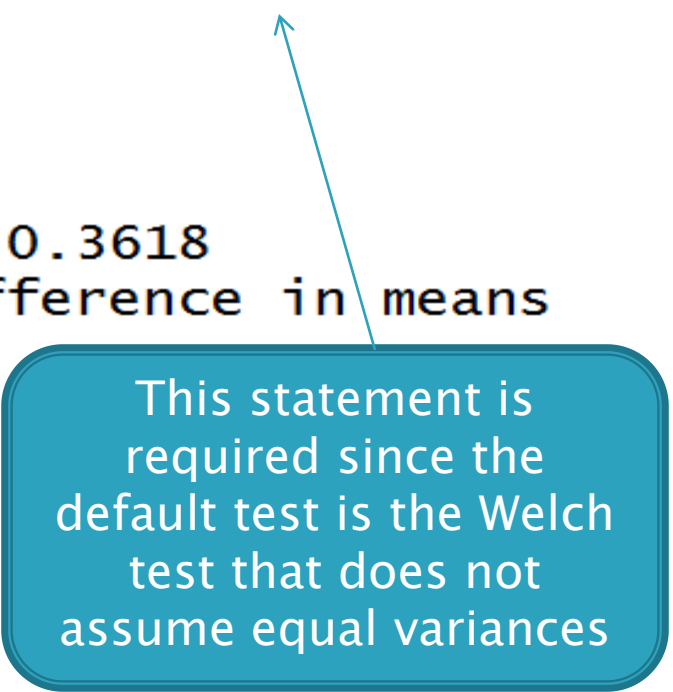
Independent Samples t -tests

- ▶ Hypothesis #4: Is there a difference between males and females on pretest perfectionism?
- ▶ In this case we would likely want to use a t -test

```
> t.test(perf1~sex, data=dat, var.equal=TRUE)
```

```
Two Sample t-test
```

```
data:  perf1 by sex  
t = -0.9167, df = 88, p-value = 0.3618  
alternative hypothesis: true difference in means  
is not equal to 0  
95 percent confidence interval:  
 -6.034613  2.224762  
sample estimates:  
mean in group f mean in group m  
    79.47360      81.37852
```



This statement is required since the default test is the Welch test that does not assume equal variances

Independent Samples t -tests

- ▶ We can also run a t -test by specifying the groups to be compared

```
> t.test(perf1[sex == "m"],perf1[sex == "f"], var  
.equal=TRUE, data=dat)
```

Two sample t-test

```
data:  perf1[sex == "m"] and perf1[sex == "f"]  
t = 0.9167, df = 88, p-value =  
0.3618  
alternative hypothesis: true difference in means  
is not equal to 0  
95 percent confidence interval:  
 -2.224762  6.034613  
sample estimates:  
mean of x mean of y  
 81.37852  79.47360
```

Independent Samples t -tests Under Variance Heterogeneity

- ▶ Welch's two independent samples t -test
 - This is the default since this a better overall test than the traditional t -test

```
> t.test(perf1~sex, data=dat)
```

```
welch Two Sample t-test
```

```
data:  perf1 by sex
t = -0.9075, df = 80.079,
p-value = 0.3669
alternative hypothesis: true difference
in means is not equal to 0
95 percent confidence interval:
 -6.082220  2.272369
sample estimates:
mean in group f mean in group m
      79.47360      81.37852
```

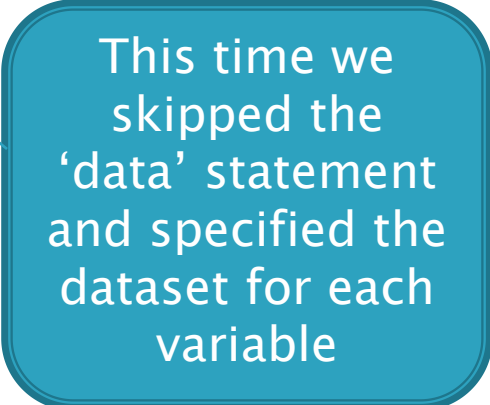
Independent Samples t -tests Under Nonnormality

- ▶ Wilcoxon–Mann–Whitney nonparametric two independent samples test

```
> wilcox.test(dat$perf1~dat$sex)
```

```
wilcoxon rank sum test with  
continuity correction
```

```
data: dat$perf1 by dat$sex  
W = 905, p-value = 0.4429  
alternative hypothesis: true location  
shift is not equal to 0
```



This time we skipped the 'data' statement and specified the dataset for each variable

What about a t -test for Nonnormality and Variance Inequality?

- ▶ Several procedures have been proposed, although the Welch t -test on trimmed means has garnered the most attention
- ▶ Rand Wilcox has written functions that accompany his texts on robust statistics that includes a function for computing the trimmed Welch t (which was developed by Yuen and often referred to as the Yuen test)
 - We will use a package called 'WRS2' that makes available many of Wilcox's most popular functions

Working with Trimmed Means

- ▶ Since we will be testing the null hypothesis that the population trimmed means are equal, it is informative to look at the trimmed means

```
> mean(dat$perf1[dat$sex=='m'], tr=.2)
```

```
[1] 80.44451
```

```
> tapply(dat$perf1, dat$sex, mean, tr=.2)
```

```
      f      m
```

```
79.04677 80.44451
```

Yuen Test

- ▶ Note that like many of Wilcox's functions, the output is not very fancy

```
> yuen(dat$perf1~dat$sex)
```

```
call:
```

```
yuen(formula = dat$perf1 ~ dat$sex)
```

```
Test statistic: 0.7588, p-value = 0.45194
```

```
95 percent confidence interval:
```

```
-5.108      2.3125
```

Population trimmed means do not differ significantly

Paired Samples t-tests

- ▶ Hypothesis #5: Is there a difference between pre and post perfectionism scores?

```
> t.test(dat$perf1, dat$perf3, paired=TRUE)
```

```
Paired t-test
```

```
data: dat$perf1 and dat$perf3  
t = 8.8712, df = 89, p-value =  
6.942e-14
```

```
alternative hypothesis: true difference in  
means is not equal to 0
```

```
95 percent confidence interval:
```

```
8.557716 13.497733
```

```
sample estimates:
```

```
mean of the differences  
11.02772
```

Paired Samples under Nonnormality

- ▶ If the distribution of difference scores is not normally distributed, the Wilcoxon signed ranks test can be much more powerful than the paired samples t-test

```
> wilcox.test(dat$perf1, dat$perf3, paired=TRUE)
```

```
wilcoxon signed rank test with  
continuity correction
```

```
data: dat$perf1 and dat$perf3  
V = 3676, p-value = 5.732e-11  
alternative hypothesis: true location shift is not  
equal to 0
```