Part I

Syllabus and Software

Part II

Summaries and displays, and one-, two-, and many-way tests of means

Part III

Nonparametric, categorical, and regression methods

Chapter 6

Nonparametric Methods

Contents

Learning objectives

After completing this topic, you should be able to:

select the appropriate procedure based on assumptions.

explain reason for using one procedure over another.

decide whether the medians between multiple populations are different.

Achieving these goals contributes to mastery in these course learning outcomes:

3. select correct statistical procedure.

5. define parameters of interest and hypotheses in words and notation.

6. summarize data visually, numerically, and descriptively.

8. use statistical software.

10. identify and explain statistical methods, assumptions, and limitations.

12. make evidence-based decisions.

6.1 Introduction

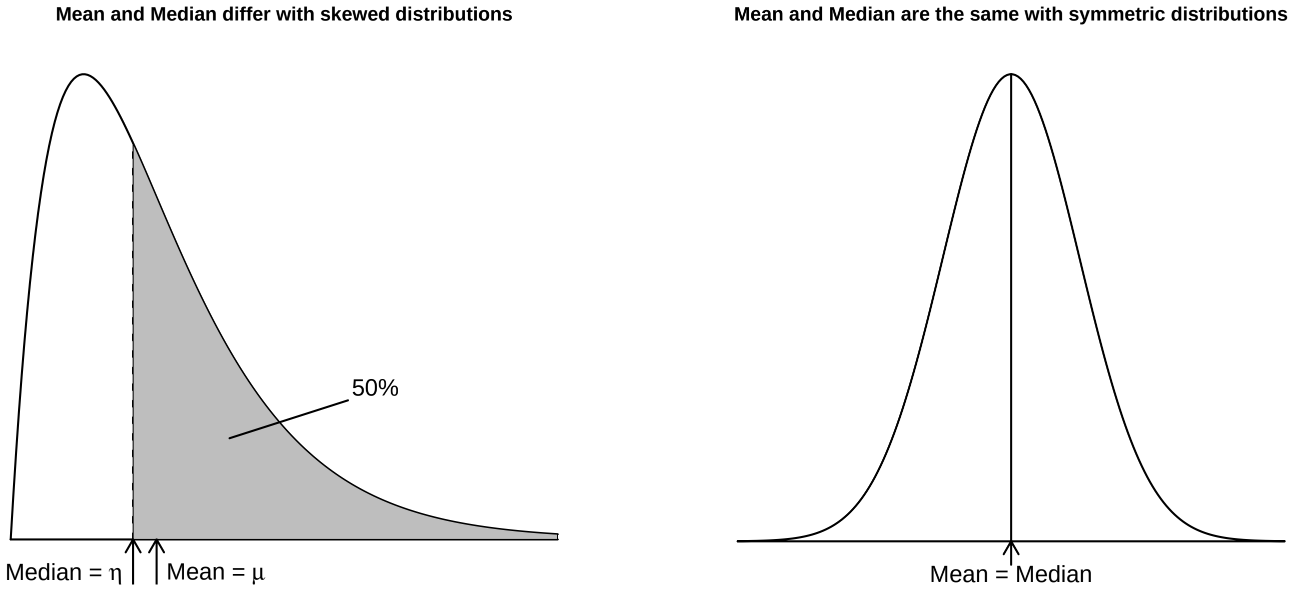
Nonparametric methods do not require the normality assumption of classical techniques. When the normality assumption is met, the ANOVA and t-test are most powerful, in that if the alternative is true these methods will make the correct decision with highest probability. However, if the normality assumption is not met, results from the ANOVA and t-test can be misleading and too liberal. I will describe and illustrate selected **non-parametric methods**, and compare them with classical methods. Some motivation and discussion of the strengths and weaknesses of non-parametric methods is given.

6.2 The Sign Test and CI for a Population Median

The **sign test** assumes that you have a random sample from a population, but makes **no assumption** about the population shape. The standard t-test provides inferences on a population mean. The sign test, in contrast, provides inferences about a **population median**.

If the population frequency curve is symmetric (see below), then the population median, identified by η, and the population mean μ are identical. In this case the sign procedures provide inferences for the population mean, though less powerfully than the t-test.

The idea behind the sign test is straightforward. Suppose you have a sample of size m from the population, and you wish to test $H_0 : \eta = \eta_0$ (a given value). Let S be the number of sampled observations *above* η_0. If H_0 is true, you expect S to be approximately one-half the sample size, $0.5m$. If S is much greater than $0.5m$, the data suggests that $\eta > \eta_0$. If S is much less than $0.5m$, the data suggests that $\eta < \eta_0$.



S has a **Binomial distribution** when H_0 is true. The Binomial distribution is used to construct a test with size α (approximately). For a two-sided alternative $H_A : \eta \neq \eta_0$, the test rejects H_0 when S is significantly different from $0.5m$, as determined from the reference Binomial distribution. One-sided tests use the corresponding lower or upper tail of the distribution. To generate a CI for η, you can exploit the duality between CI and tests. A $100(1-\alpha)\%$ CI for η consists of all values η_0 not rejected by a two-sided size α test of $H_0 : \eta = \eta_0$.

Not all test sizes and confidence levels are possible because the test statistic S is discrete valued. R's `SIGN.test()` in the `BSDA` package gives an exact p-value for the test, and approximates the desired confidence level using a linear interpolation algorithm.

Example: Income Data Recall that the income distribution is extremely skewed, with two extreme outliers at 46 and 1110.

```
#### Example: Income Data
income <- c(7, 1110, 7, 5, 8, 12, 0, 5, 2, 2, 46, 7)
# sort in decreasing order
income <- sort(income, decreasing = TRUE)
income
```

```
##  [1] 1110   46   12    8    7    7    7    5    5    2    2    0
```

```
summary(income)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.00    4.25    7.00  100.92    9.00 1110.00
```

```
sd(income)
```

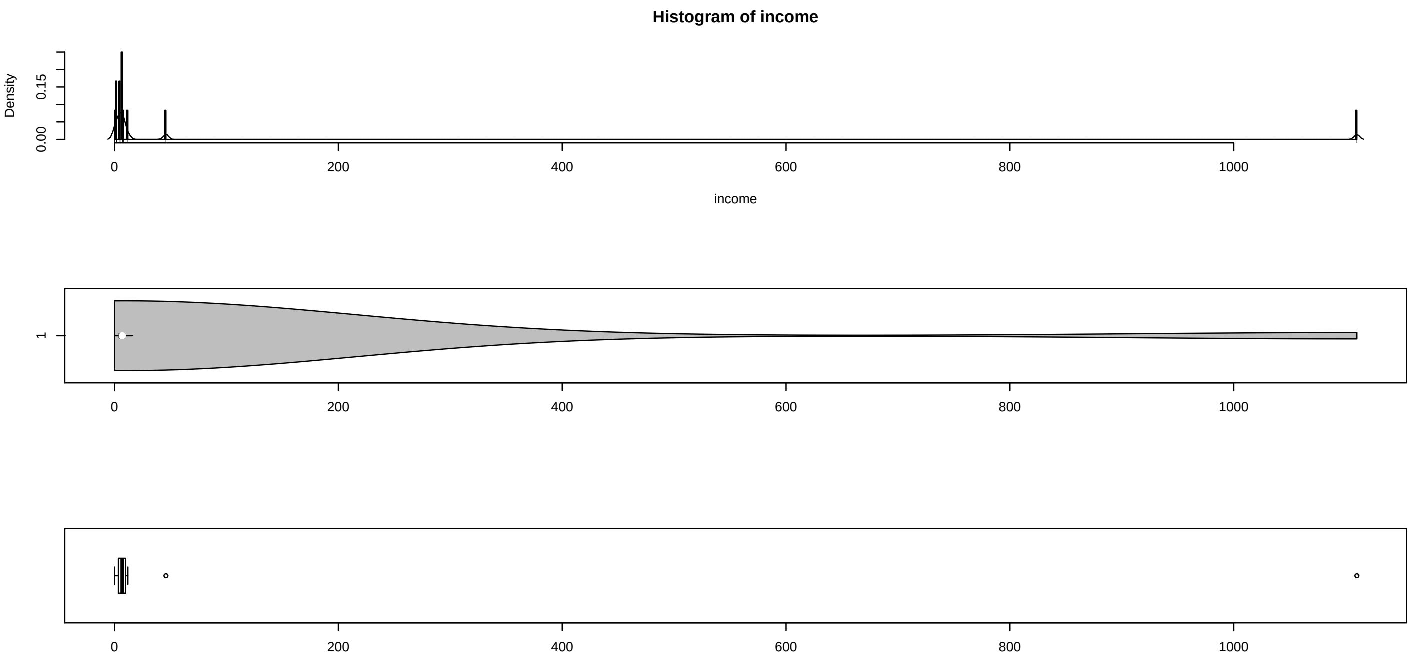
```
## [1] 318.0078
```

The income data is unimodal, skewed right, with two extreme outliers.

```
par(mfrow=c(3,1))
# Histogram overlaid with kernel density curve
hist(income, freq = FALSE, breaks = 1000)
points(density(income), type = "l")
rug(income)

# violin plot
library(vioplot)
vioplot(income, horizontal=TRUE, col="gray")
```

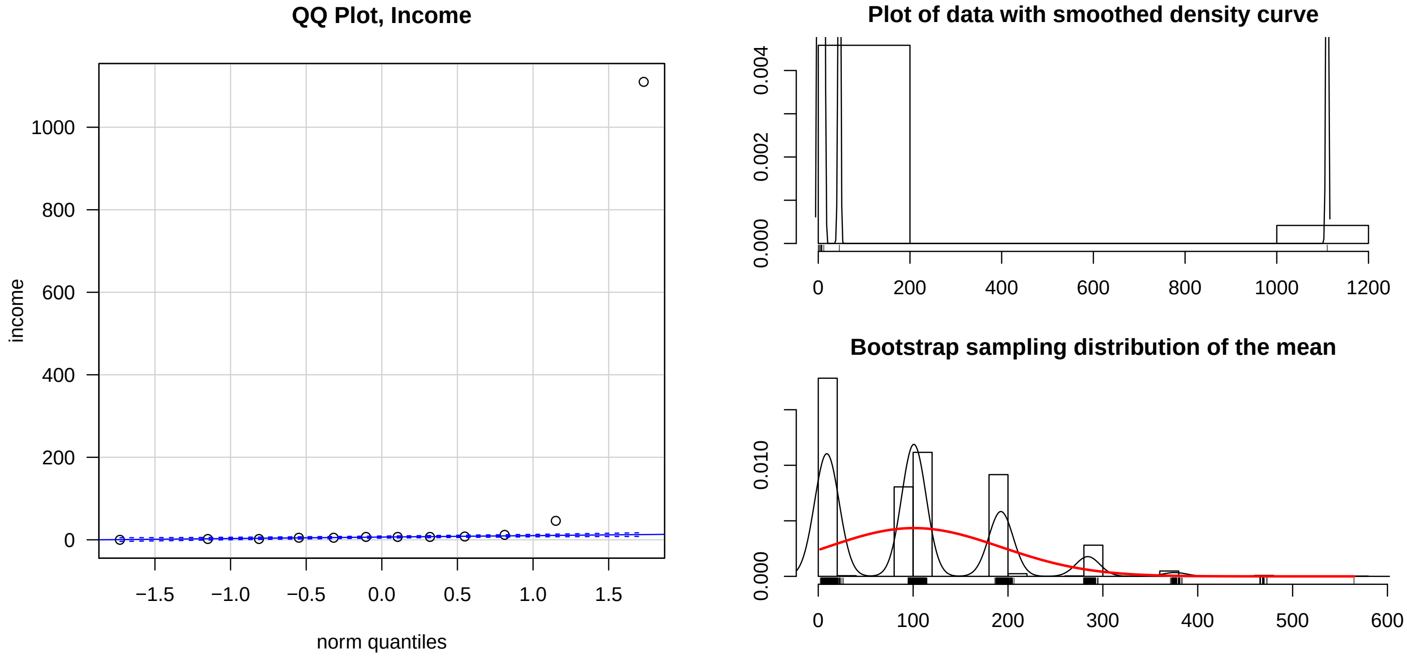
```
## [1]    0 1110
```

```
# boxplot
boxplot(income, horizontal=TRUE)
```



The normal QQ-plot of the sample data indicates strong deviation from normality, and the CLT can't save us: even the bootstrap sampling distribution of the mean indicates strong deviation from normality.

```r
library(car)
qqPlot(income, las = 1, id = list(n = 0, cex = 1), lwd = 1, main="QQ Plot, Income")

bs.one.samp.dist(income)
```



The presence of the outliers has a dramatic effect on the 95% CI for the population mean income μ, which goes from -101 to 303 (in 1000 dollar units).

This t-CI is suspect because the normality assumption is unreasonable. A CI for the population median income η is more sensible because the median is likely to be a more reasonable measure of typical value. Using the sign procedure, you are 95% confident that the population median income is between 2.32 and 11.57 (times $1000).

```
library(BSDA)
t.test(income)
```

```
##
##  One Sample t-test
##
## data:  income
## t = 1.0993, df = 11, p-value = 0.2951
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -101.1359  302.9692
## sample estimates:
## mean of x
##  100.9167
```

```
SIGN.test(income)
```

```
##
##  One-sample Sign-Test
##
## data:  income
## s = 11, p-value = 0.0009766
## alternative hypothesis: true median is not equal to 0
## 95 percent confidence interval:
##   2.319091 11.574545
## sample estimates:
## median of x
##           7
##
## Achieved and Interpolated Confidence Intervals:
##
##                   Conf.Level L.E.pt  U.E.pt
## Lower Achieved CI     0.8540 5.0000  8.0000
## Interpolated CI       0.9500 2.3191 11.5745
## Upper Achieved CI     0.9614 2.0000 12.0000
```

Example: Age at First Heart Transplant Recall that the distribution of ages is skewed to the left with a lower outlier. A question of interest is whether the "typical age" at first transplant is 50. This can be formulated as a test about the population median η or as a test about the population mean μ, depending on the interpretation.

```
#### Example: Age at First Heart Transplant
age <- c(54, 42, 51, 54, 49, 56, 33, 58, 54, 64, 49)
# sort in decreasing order
age <- sort(age, decreasing = TRUE)
age
```

```
##  [1] 64 58 56 54 54 54 51 49 49 42 33
```

```
summary(age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.00   49.00   54.00   51.27   55.00   64.00
```

```
sd(age)
```

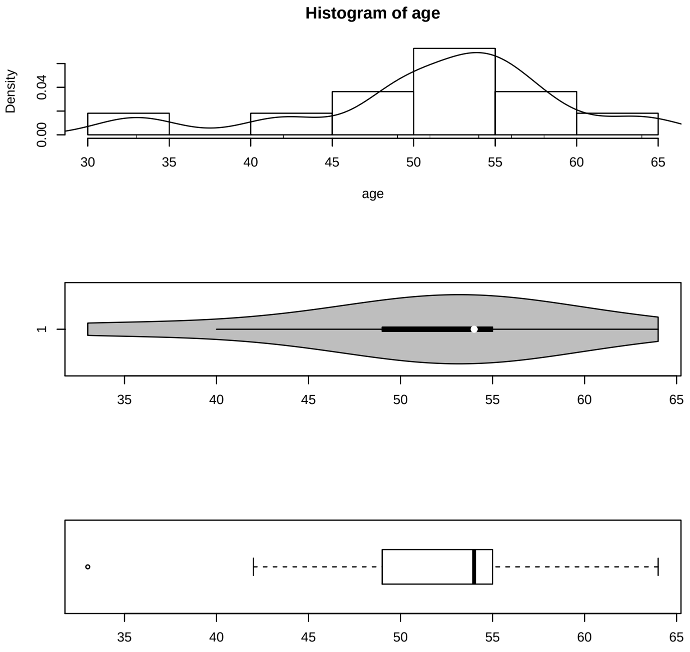
```
## [1] 8.25943
```

The age data is unimodal, skewed left, no extreme outliers.

```
par(mfrow=c(3,1))
# Histogram overlaid with kernel density curve
hist(age, freq = FALSE, breaks = 10)
points(density(age), type = "l")
rug(age)

# violin plot
library(vioplot)
vioplot(age, horizontal=TRUE, col="gray")
```

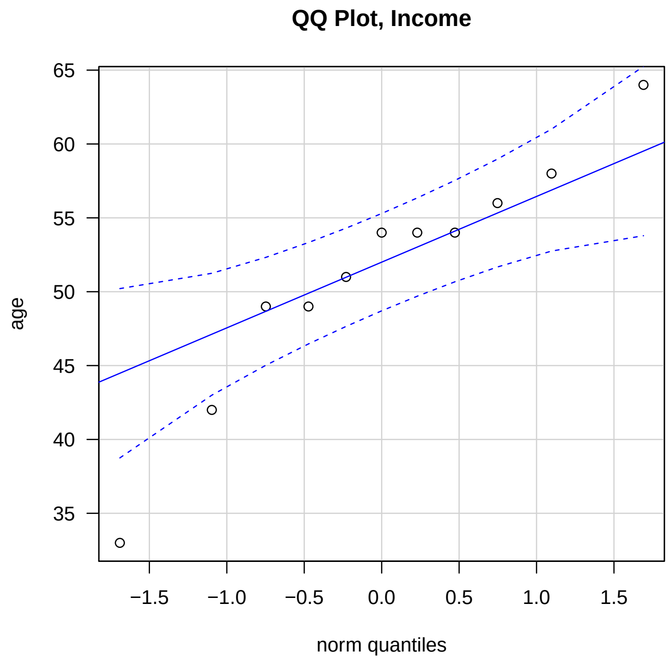
```
## [1] 33 64
```

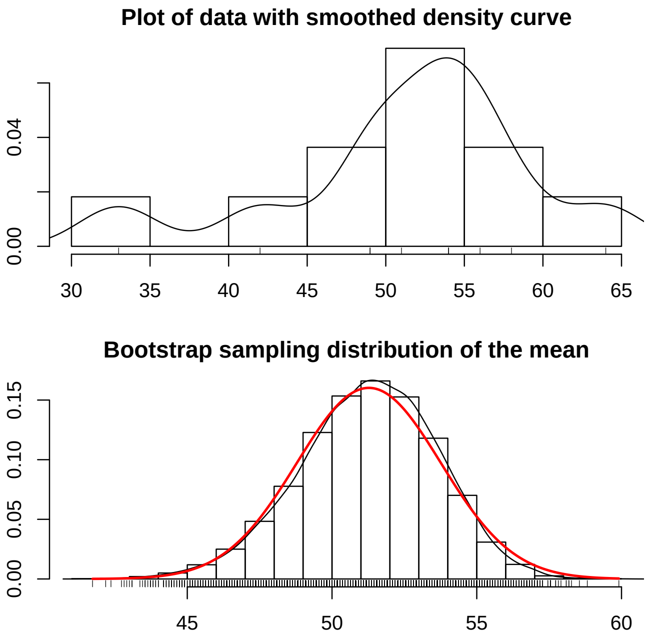
```
# boxplot
boxplot(age, horizontal=TRUE)
```



The normal QQ-plot of the sample data indicates mild deviation from normality in the left tail (2 points of 11 outside the bands), and the bootstrap sampling distribution of the mean indicates weak deviation from normality. It is good practice in this case to use the nonparametric test as a double-check of the *t*-test, with the nonparametric test being the more conservative test.

```
library(car)
qqPlot(age, las = 1, id = list(n = 0, cex = 1), lwd = 1, main="QQ Plot, Income")

bs.one.samp.dist(age)
```





The sign test for $H_0 : \eta = 50$ against $H_A : \eta \neq 50$ has a p-value of 0.549, which is not sufficient to reject H_0. A 95% CI for η is 47.0 to 56.6 years, which includes the hypothesized median age of 50. Similar conclusions are reached with the *t*-CI and the test on μ, but you should have less confidence in these results because the normality assumption is tenuous.

```
library(BSDA)
t.test(age, mu=50)
##
##  One Sample t-test
##
## data:  age
## t = 0.51107, df = 10, p-value = 0.6204
## alternative hypothesis: true mean is not equal to 50
## 95 percent confidence interval:
##  45.72397 56.82149
## sample estimates:
## mean of x
##  51.27273
SIGN.test(age, md=50)
##
```

```
##  One-sample Sign-Test
##
## data:  age
## s = 7, p-value = 0.5488
## alternative hypothesis: true median is not equal to 50
## 95 percent confidence interval:
##  46.98909 56.57455
## sample estimates:
## median of x
##          54
##
## Achieved and Interpolated Confidence Intervals:
##
##                    Conf.Level  L.E.pt  U.E.pt
## Lower Achieved CI     0.9346 49.0000 56.0000
## Interpolated CI       0.9500 46.9891 56.5745
## Upper Achieved CI     0.9883 42.0000 58.0000
```

6.3 Wilcoxon Signed-Rank Procedures

The **Wilcoxon** procedure assumes you have a random sample from a population with a **symmetric** frequency curve. The curve need not be normal. The test and CI can be viewed as procedures for either the population median or mean.

To illustrate the computation of the Wilcoxon statistic W, suppose you wish to test $H_0 : \mu = \mu_0 = 10$ on the made-up data below. The test statistic requires us to compute the **signs** of $X_i - \mu_0$ and the **ranks** of $|X_i - \mu_0|$. Ties in $|X_i - \mu_0|$ get the average rank and observations at μ_0 (here 10) are always discarded. The Wilcoxon statistic is the **sum of the signed ranks for observations above** $\mu_0 = 10$. For us

$$W \;=\; 6 + 4.5 + 8 + 2 + 4.5 + 7 \;=\; 32.$$

X_i	$X_i - 10$	sign	$\lvert X_i - 10\rvert$	rank	sign× rank
20	10	+	10	6	6
18	8	+	8	4.5	4.5
23	13	+	13	8	8
5	−5	−	5	3	−3
14	4	+	4	2	2
8	−2	−	2	1	−1
18	8	+	8	4.5	4.5
22	12	+	12	7	7

The sum of all ranks is always $0.5m(m+1)$, where m is the sample size. If H_0 is true, you expect W to be approximately $0.5 \times 0.5m(m+1) = 0.25m(m+1)$. Why? Recall that W adds up the ranks for observations above μ_0. If H_0 is true, you expect 1/2 of all observations to be above μ_0, assuming the population distribution is symmetric. The ranks of observations above μ_0 should add to approximately 1/2 times the sum of all ranks. You reject H_0 in favor of $H_A : \mu \neq \mu_0$ if W is much larger than, or much smaller than $0.25m(m+1)$. One sided tests can also be constructed. The Wilcoxon CI for μ is computed in a manner analogous to that described for the sign CI.

Here, $m = 8$ so the sum of all ranks is $0.5 \times 8 \times 9 = 36$ (check yourself). The expected value of W is $0.5 \times 0.5 \times 8 \times 9 = 18$. Is the observed value of $W = 32$ **far from** the expected value of 18? To formally answer this question, we need to use the Wilcoxon procedures, which are implemented in R with `wilcox.test()`.

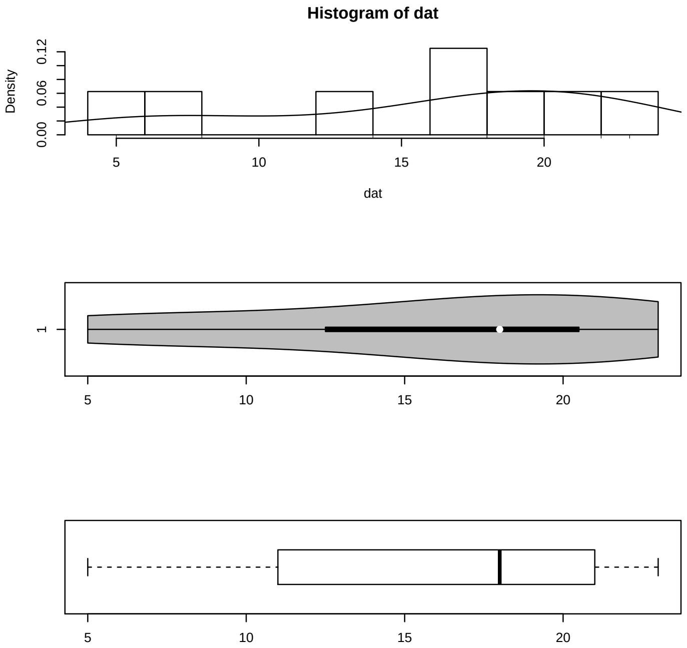
Example: Made-up Data The boxplot indicates that the distribution is fairly symmetric, so the Wilcoxon method is reasonable (so is a *t*-CI and test).

```
#### Example: Made-up Data
dat <- c(20, 18, 23,  5, 14,  8, 18, 22)
# sort in decreasing order
dat <- sort(dat, decreasing = TRUE)
dat

## [1] 23 22 20 18 18 14  8  5

summary(dat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.0    12.5    18.0    16.0    20.5    23.0
sd(dat)
## [1] 6.524678
```

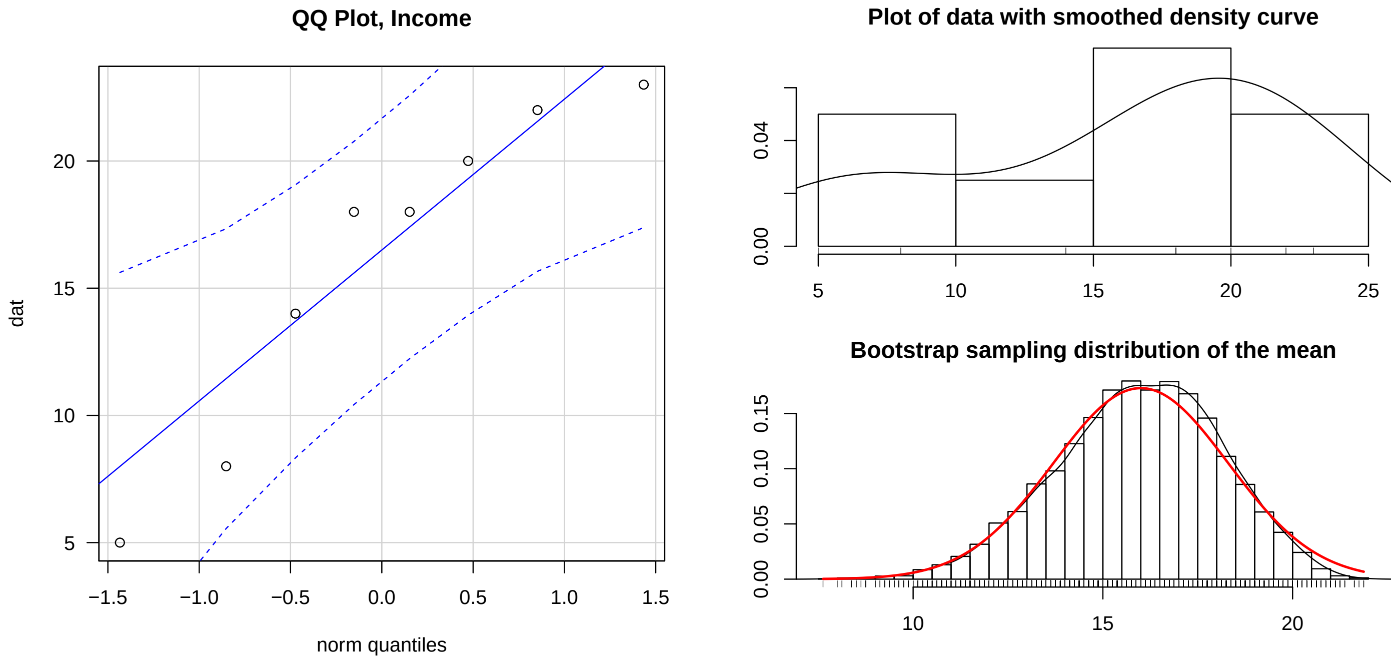
The dat data is unimodal, skewed left, no extreme outliers.

```
par(mfrow=c(3,1))
# Histogram overlaid with kernel density curve
hist(dat, freq = FALSE, breaks = 10)
points(density(dat), type = "l")
rug(dat)

# violin plot
library(vioplot)
vioplot(dat, horizontal=TRUE, col="gray")
## [1]  5 23
# boxplot
boxplot(dat, horizontal=TRUE)
```



The normal QQ-plot of the sample data indicates insufficient evidence of deviation from normality though both the QQ-plot and the bootstrap sampling distribution of the mean indicates weak left-skewness. Either the Wilcoxon or t-test are appropriate.

```
par(mfrow=c(1,1))
library(car)
qqPlot(dat, las = 1, id = list(n = 0, cex = 1), lwd = 1, main="QQ Plot, Income")

bs.one.samp.dist(dat)
```



The Wilcoxon p-value with continuity correction for testing $H_0 : \mu = 10$ against a two-sided alternative is 0.058. This would not lead to rejecting H_0 at the 5% level.

```
t.test(dat, mu=10)
##
##  One Sample t-test
##
## data:  dat
## t = 2.601, df = 7, p-value = 0.03537
## alternative hypothesis: true mean is not equal to 10
## 95 percent confidence interval:
##  10.54523 21.45477
## sample estimates:
## mean of x
##        16
# with continuity correction in the normal approximation for the p-value
wilcox.test(dat, mu=10, conf.int=TRUE)
## Warning in wilcox.test.default(dat, mu = 10, conf.int = TRUE): cannot compute exact p-value
with ties
## Warning in wilcox.test.default(dat, mu = 10, conf.int = TRUE): cannot compute exact confide
interval with ties
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat
## V = 32, p-value = 0.0584
## alternative hypothesis: true location is not equal to 10
## 95 percent confidence interval:
##   9.500002 21.499942
## sample estimates:
## (pseudo)median
##        16.0056
```

```
# without continuity correction
wilcox.test(dat, mu=10, conf.int=TRUE, correct=FALSE)
```

```
## Warning in wilcox.test.default(dat, mu = 10, conf.int = TRUE, correct = FALSE): cannot
compute exact p-value with ties
## Warning in wilcox.test.default(dat, mu = 10, conf.int = TRUE, correct = FALSE): cannot
compute exact confidence interval with ties
##
##  Wilcoxon signed rank test
##
## data:  dat
## V = 32, p-value = 0.04967
## alternative hypothesis: true location is not equal to 10
## 95 percent confidence interval:
##  10.99996 21.00005
## sample estimates:
## (pseudo)median
##        16.0056
```

6.3.1 Nonparametric Analyses of Paired Data

Nonparametric methods for single samples can be used to analyze paired data because the difference between responses within pairs is the unit of analysis.

Example: Sleep Remedies I will illustrate Wilcoxon methods on the paired comparison of two remedies A and B for insomnia. The number of hours of sleep gained on each method was recorded.

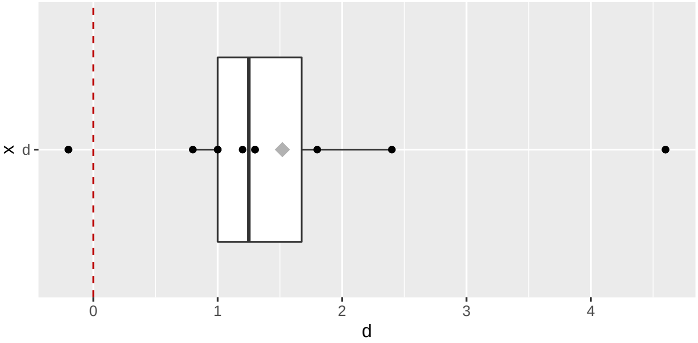
```
#### Example: Sleep Remedies
# Data and numerical summaries
a <- c( 0.7, -1.6, -0.2, -1.2,  0.1,  3.4,  3.7,  0.8,  0.0,  2.0)
b <- c( 1.9,  0.8,  1.1,  0.1, -0.1,  4.4,  5.5,  1.6,  4.6,  3.0)
d <- b - a;
sleep <- data.frame(a, b, d)
summary(sleep$d)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -0.200   1.000   1.250   1.520   1.675   4.600
```

```
shapiro.test(sleep$d)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sleep$d
## W = 0.83798, p-value = 0.04173
```

```
# boxplot
library(ggplot2)
p3 <- ggplot(sleep, aes(x = "d", y = d))
p3 <- p3 + geom_hline(yintercept=0, colour="#BB0000", linetype="dashed")
p3 <- p3 + geom_boxplot()
p3 <- p3 + geom_point()
p3 <- p3 + stat_summary(fun.y = mean, geom = "point", shape = 18,
                        size = 4, alpha = 0.3)
p3 <- p3 + coord_flip()
print(p3)
```



The boxplot shows that distribution of differences is reasonably symmetric but not normal. Recall that the Shapiro-Wilk test of normality was significant at the 5% level (p-value=0.042). It is sensible to use the Wilcoxon procedure on the differences. Let μ_B be the population mean sleep gain on remedy B, and μ_A be the population mean sleep gain on remedy A. You are 95% confident that $\mu_B - \mu_A$ is between 0.8 and 2.8 hours. Putting this another way, you are 95% confident that μ_B exceeds μ_A by between 0.8 and 2.8 hours. The p-value for testing $H_0 : \mu_B - \mu_A = 0$ against a two-sided alternative is 0.008, which strongly suggests that $\mu_B \neq \mu_A$. This agrees with the CI. Note that the t-CI and test give qualitatively similar conclusions as the Wilcoxon methods, but the t-test p-value is about half as large.

If you are uncomfortable with the symmetry assumption, you could use the sign CI for the population median difference between B and A. I will note that a 95% CI for the median difference goes from 0.86 to 2.2 hours.

```
t.test(sleep$d, mu=0)
##
##  One Sample t-test
##
## data:  sleep$d
## t = 3.7796, df = 9, p-value = 0.004352
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.610249 2.429751
## sample estimates:
## mean of x
##      1.52
# with continuity correction in the normal approximation for the p-value
wilcox.test(sleep$d, mu=0, conf.int=TRUE)
## Warning in wilcox.test.default(sleep$d, mu = 0, conf.int = TRUE): cannot compute exact
p-value with ties
## Warning in wilcox.test.default(sleep$d, mu = 0, conf.int = TRUE): cannot compute exact
confidence interval with ties
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  sleep$d
## V = 54, p-value = 0.008004
## alternative hypothesis: true location is not equal to 0
## 95 percent confidence interval:
##  0.7999339 2.7999620
## sample estimates:
## (pseudo)median
##       1.299983
# can use the paired= option
#wilcox.test(sleep£b, sleep£a, paired=TRUE, mu=0, conf.int=TRUE)
# if don't assume symmetry, can use sign test
#SIGN.test(sleep£d)
```

6.3.2 Comments on One-Sample Nonparametric Methods

For this discussion, I will assume that the underlying population distribution is (approximately) symmetric, which implies that population means and medians are equal (approximately). For symmetric distributions the t, sign, and Wilcoxon procedures are all appropriate.

If the underlying population distribution is extremely skewed, you can use the sign procedure to get a CI for the population median. Alternatively, as illustrated on HW 2, you can transform the data to a scale where the underlying distribution is nearly normal, and then use the classical t-methods. Moderate degrees of skewness will not likely have a big impact on the standard t-test and CI.

The one-sample t-test and CI are optimal when the underlying population frequency curve is normal. Essentially this means that the t-CI is, on average, narrowest among all CI procedures with given level, or that the t-test has the highest power among all tests with a given size. The width of a CI provides a measure of the sensitivity of the estimation method. For a given level CI, the narrower CI better pinpoints the unknown population mean.

With heavy-tailed symmetric distributions, the t-test and CI tend to be conservative. Thus, for example, a nominal 95% t-CI has actual coverage rates higher than 95%, and the nominal 5% t-test has an actual size smaller than 5%. The t-test and CI possess a property that is commonly called **robustness of validity**. However, data from heavy-tailed distributions can have a profound effect on the **sensitivity** of the t-test and CI. Outliers can dramatically inflate the standard error of the mean, causing the CI to be needlessly wide, and tests to have diminished power (outliers typically inflate p-values for the t-test). The sign and Wilcoxon procedures downweight the influence of outliers by looking at sign or signed-ranks instead of the actual data values. These two nonparametric methods are somewhat less efficient than the t-methods when the population is normal (efficiency is about 0.64 and 0.96 for the sign and Wilcoxon methods relative to the normal t-methods, where efficiency is the ratio of sample sizes needed for equal power), but can be infinitely more efficient with heavier than normal tailed distributions. In essence, the t-methods do not have a **robustness of sensitivity**.

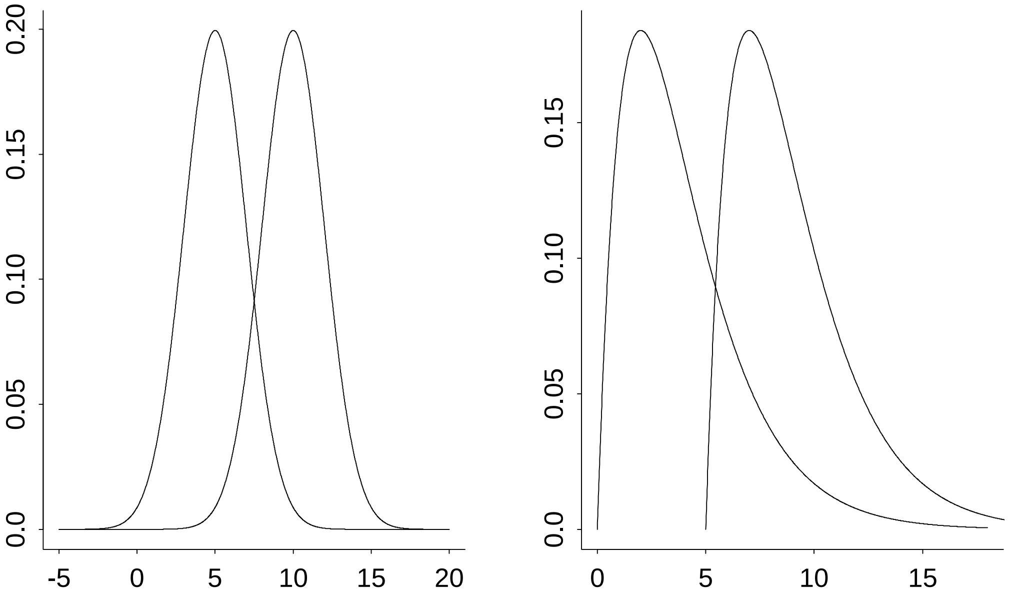
Nonparametric methods have gained widespread acceptance in many scientific disciplines, but not all. Scientists in some disciplines continue to use classical t-methods because they believe that the methods are robust to non-normality. As noted above, this is a robustness of validity, not sensitivity. This misconception is unfortunate, and results in the routine use of methods that are less powerful than the non-parametric techniques. **Scientists need to be flexible and adapt their tools to the problem at hand, rather than use the same tool indiscriminately!** I have run into suspicion that use of nonparametric methods was an attempt to "cheat" in some way — properly applied, they are excellent tools that *should* be used.

A minor weakness of nonparametric methods is that they do not easily generalize to complex modelling problems. A great deal of progress has been made in this area, but most software packages have not included the more advanced techniques (R is among the forerunners).

Nonparametric statistics used to refer almost exclusively to the set of methods such as we have been discussing that provided analogs like tests and CIs to the normal theory methods without requiring the assumption of sampling from normal distributions. There is now a large area of statistics also called nonparametric methods not focused on these goals at all. In our department we (used to) have a course titled "Nonparametric Curve Estimation & Image Reconstruction", where the focus is much more general than relaxing an assumption of normality. In that sense, what we are covering in this course could be considered "classical" nonparametrics.

6.4 (Wilcoxon-)Mann-Whitney Two-Sample Procedure

The WMW procedure assumes you have independent random samples from the two populations, and assumes that the populations have the **same shapes and spreads** (the frequency curves for the two populations are "shifted" versions of each other — see below). The frequency curves are not required to be symmetric. The WMW procedures give a CI and tests on the difference $\eta_1 - \eta_2$ between the two population medians. If the populations are symmetric, then the methods apply to $\mu_1 - \mu_2$.



The R help on `?wilcox.test` gives references to how the exact WMW procedure is actually calculated; here is a good approximation to the exact method that is easier to understand. The WMW procedure is based on ranks. The two samples are combined, ranked from smallest to largest (1=smallest) and separated back into the original samples. If the two populations have equal medians, you expect the average rank in the two samples to be roughly equal. The WMW test computes a classical two sample t-test using the pooled variance on the ranks to assess whether the sample mean ranks are significantly different.

Example: Comparison of Cooling Rates of Uwet and Walker Co. Meteorites The Uwet[1] (Cross River, Nigeria) and Walker[2] County (Alabama, US) meteorite cooling rate data are below. A primary interest is comparing the population "typical" cooling rate measurements.

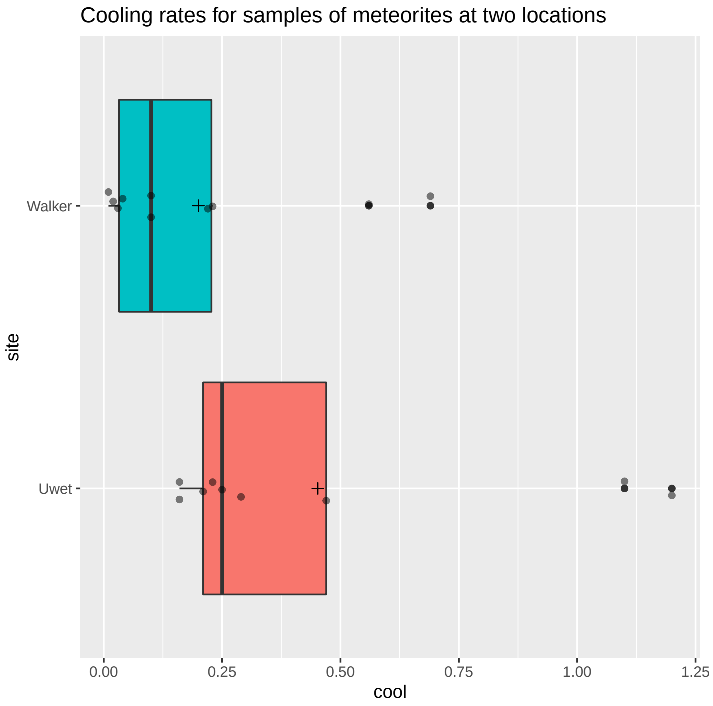
[1] http://www.lpi.usra.edu/meteor/metbull.php?code=24138

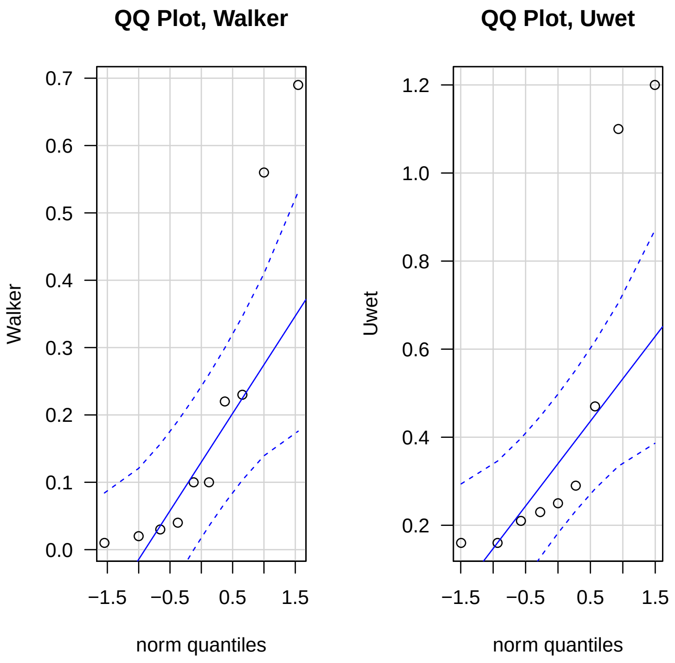
[2] http://www.lpi.usra.edu/meteor/metbull.php?code=24204

```
#### Example: Comparison of Cooling Rates of Uwet and Walker Co. Meteorites
Uwet   <- c(0.21, 0.25, 0.16, 0.23, 0.47, 1.20, 0.29, 1.10, 0.16)
Walker <- c(0.69, 0.23, 0.10, 0.03, 0.56, 0.10, 0.01, 0.02, 0.04, 0.22)
```

The boxplots and normal QQ-plots show that the distributions are rather skewed to the right. The AD test of normality indicate that a normality assumption is unreasonable for each population.

```
met <- data.frame(Uwet=c(Uwet,NA), Walker)
library(reshape2)
met.long <- melt(met, variable.name = "site", value.name = "cool", na.rm=TRUE)

## No id variables; using all as measure variables

# naming variables manually, the variable.name and value.name not working 11/2012
names(met.long) <- c("site", "cool")
library(ggplot2)
p <- ggplot(met.long, aes(x = site, y = cool, fill=site))
p <- p + geom_boxplot()
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0), alpha = 0.5)
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 3, size = 2)
p <- p + coord_flip()
p <- p + labs(title = "Cooling rates for samples of meteorites at two locations")
p <- p + theme(legend.position="none")
print(p)

par(mfrow=c(1,2))
library(car)
qqPlot(Walker, las = 1, id = list(n = 0, cex = 1), lwd = 1, main="QQ Plot, Walker")
qqPlot(Uwet, las = 1, id = list(n = 0, cex = 1), lwd = 1, main="QQ Plot, Uwet")
```





I carried out the standard two-sample procedures to see what happens. The pooled-variance and Satterthwaithe results are comparable, which is expected because the sample standard deviations and sample sizes are roughly equal. Both tests indicate that the mean cooling rates for Uwet and Walker Co. meteorites are not significantly different at the 10% level. You are 95% confident that the mean cooling rate for Uwet is at most 0.1 less, and no more than 0.6 greater than that for Walker Co. (in degrees per million years).

```
# numerical summaries
summary(Uwet)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1600  0.2100  0.2500  0.4522  0.4700  1.2000

c(sd(Uwet), IQR(Uwet), length(Uwet))

## [1] 0.4069944 0.2600000 9.0000000

summary(Walker)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0100  0.0325  0.1000  0.2000  0.2275  0.6900

c(sd(Walker), IQR(Walker), length(Walker))

## [1]  0.2389793  0.1950000 10.0000000

t.test(Uwet, Walker, var.equal = TRUE)

##
##  Two Sample t-test
##
## data:  Uwet and Walker
## t = 1.6689, df = 17, p-value = 0.1134
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0666266  0.5710710
## sample estimates:
## mean of x mean of y
## 0.4522222 0.2000000

t.test(Uwet, Walker)

##
##  Welch Two Sample t-test
##
## data:  Uwet and Walker
## t = 1.6242, df = 12.652, p-value = 0.129
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08420858  0.58865302
## sample estimates:
## mean of x mean of y
## 0.4522222 0.2000000
```

Given the marked skewness, a nonparametric procedure is more appropriate. The Wilcoxon-Mann-Whitney comparison of population medians is reasonable. Why? The WMW test of equal population medians is significant (barely) at the 5% level. You are 95% confident that median cooling rate for Uwet exceeds that for Walker by between 0+ and 0.45 degrees per million years.

```
wilcox.test(Uwet, Walker, conf.int = TRUE)
## Warning in wilcox.test.default(Uwet, Walker, conf.int = TRUE): cannot compute exact p-value
with ties
## Warning in wilcox.test.default(Uwet, Walker, conf.int = TRUE): cannot compute exact confide
intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Uwet and Walker
## W = 69.5, p-value = 0.04974
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  0.0000449737 0.4499654518
## sample estimates:
## difference in location
##              0.1702657
```

The difference between the WMW and t-test p-values and CI lengths (i.e. the WMW CI is narrower and the p-value smaller) reflects the effect of the outliers on the sensitivity of the standard tests and CI.

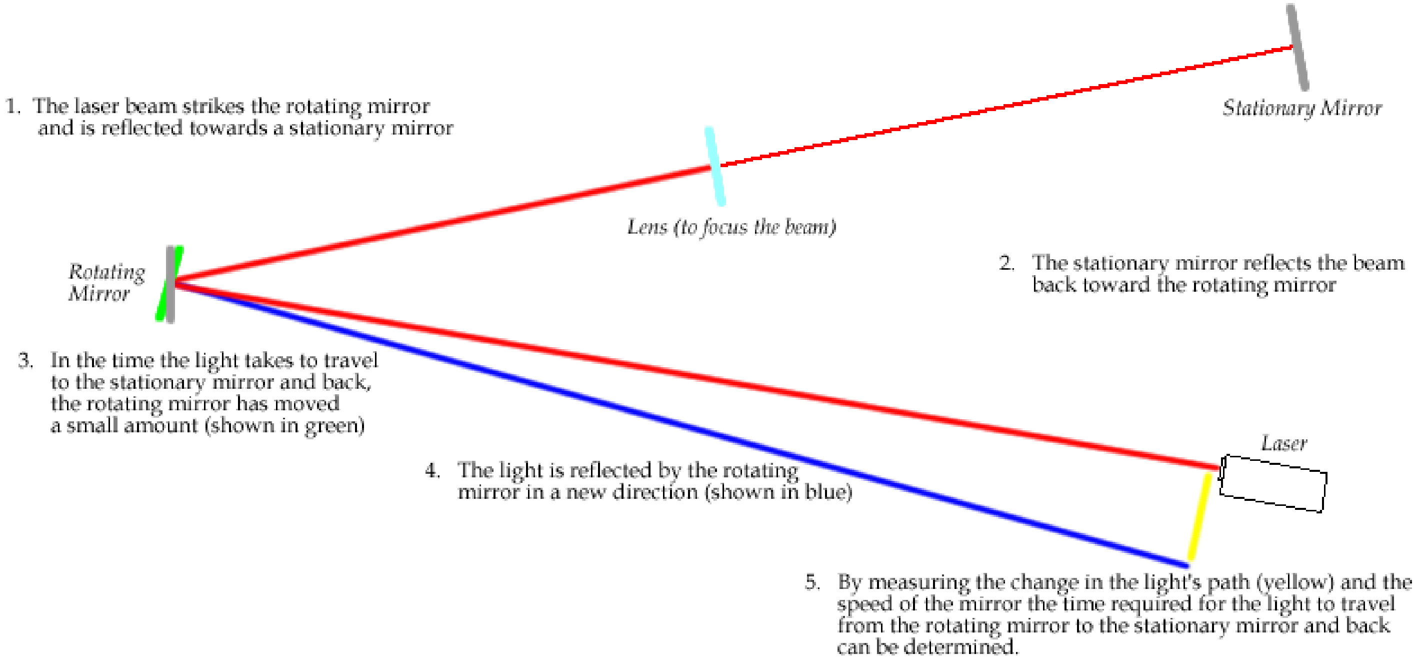
I conducted a pooled-variance two-sample t-test on ranks to show you that the p-value is close to the WMW p-value, as expected.

```
rank(met.long$cool)
##  [1]  9.0 13.0  7.5 11.5 15.0 19.0 14.0 18.0  7.5 17.0 11.5  5.5  3.0
## [14] 16.0  5.5  1.0  2.0  4.0 10.0
by(rank(met.long$cool), met.long$site, summary)
## met.long$site: Uwet
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.50    9.00   13.00   12.72   15.00   19.00
## ----------------------------------------------------
## met.long$site: Walker
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    3.25    5.50    7.55   11.12   17.00
# note: the CI for ranks is not interpretable
t.test(rank(met.long$cool) ~ met.long$site, var.equal = TRUE)
##
##  Two Sample t-test
```

```
##
## data:  rank(met.long$cool) by met.long$site
## t = 2.2082, df = 17, p-value = 0.04125
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   0.2304938 10.1139507
## sample estimates:
##   mean in group Uwet mean in group Walker
##             12.72222             7.55000
```

Example: Newcombe's Data Experiments of historical importance were performed beginning in the eighteenth century to determine physical constants, such as the mean density of the earth, the distance from the earth to the sun, and the velocity of light. An interesting series of experiments to determine the velocity of light was begun in 1875. The first method used, and reused with refinements several times thereafter, was the rotating mirror method[3]. In this method a beam of light is reflected off a rapidly rotating mirror to a fixed mirror at a carefully measured distance from the source. The returning light is re-reflected from the rotating mirror at a different angle, because the mirror has turned slightly during the passage of the corresponding light pulses. From the speed of rotation of the mirror and from careful measurements of the angular difference between the outward-bound and returning light beams, the passage time of light can be calculated for the given distance. After averaging several calculations and applying various corrections, the experimenter can combine mean passage time and distance for a determination of the velocity of light. Simon Newcombe, a distinguished American scientist, used this method during the year 1882 to generate the passage time measurements given below, in microseconds. The travel path for this experiment was 3721 meters in length, extending from Ft. Meyer, on the west bank of the Potomac River in Washington, D.C., to a fixed mirror at the base of the Washington Monument.

[3] http://en.wikipedia.org/wiki/File:Speed_of_light_(foucault).PNG

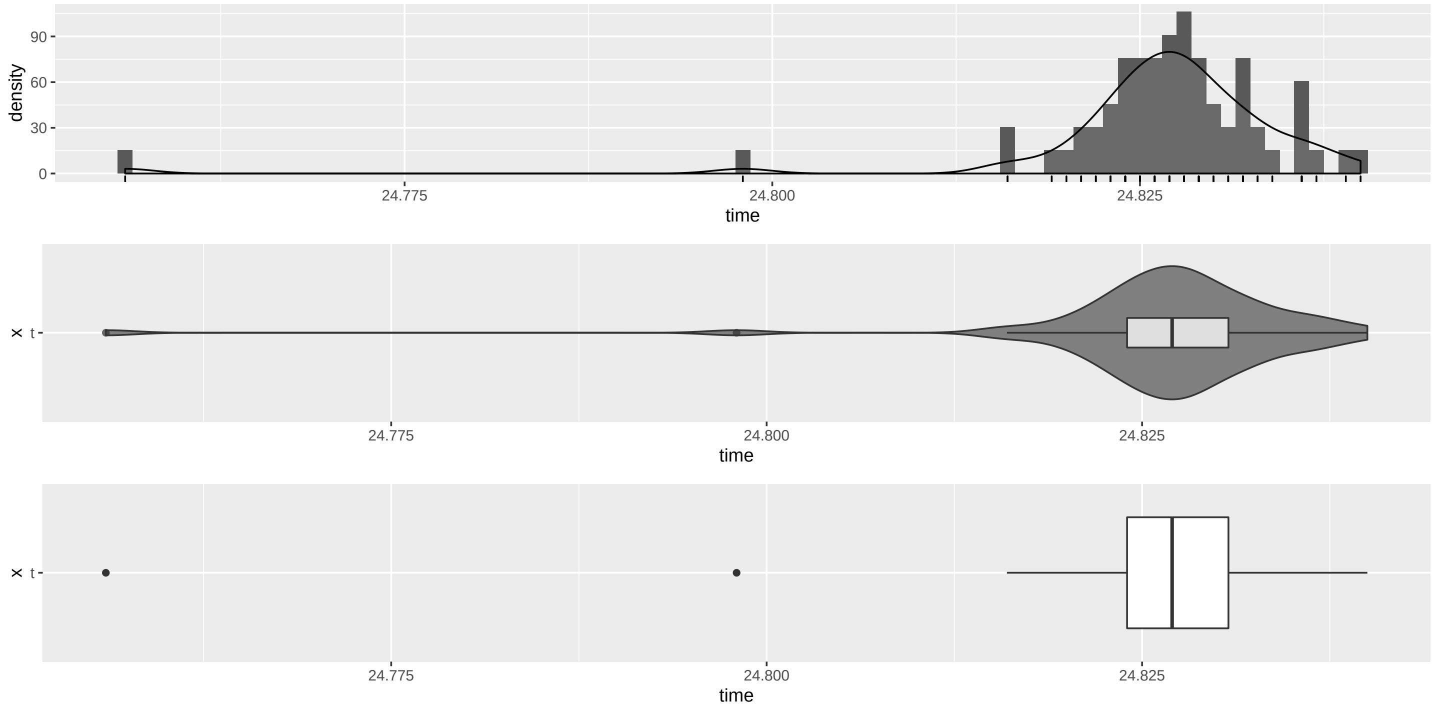

1. The laser beam strikes the rotating mirror and is reflected towards a stationary mirror
Stationary Mirror
Lens (to focus the beam)
Rotating Mirror
2. The stationary mirror reflects the beam back toward the rotating mirror
3. In the time the light takes to travel to the stationary mirror and back, the rotating mirror has moved a small amount (shown in green)
Laser
4. The light is reflected by the rotating mirror in a new direction (shown in blue)
5. By measuring the change in the light's path (yellow) and the speed of the mirror the time required for the light to travel from the rotating mirror to the stationary mirror and back can be determined.

The problem is to determine a 95% CI for the "true" passage time, which is taken to be the typical time (mean or median) of the population of measurements that were or could have been taken by this experiment.

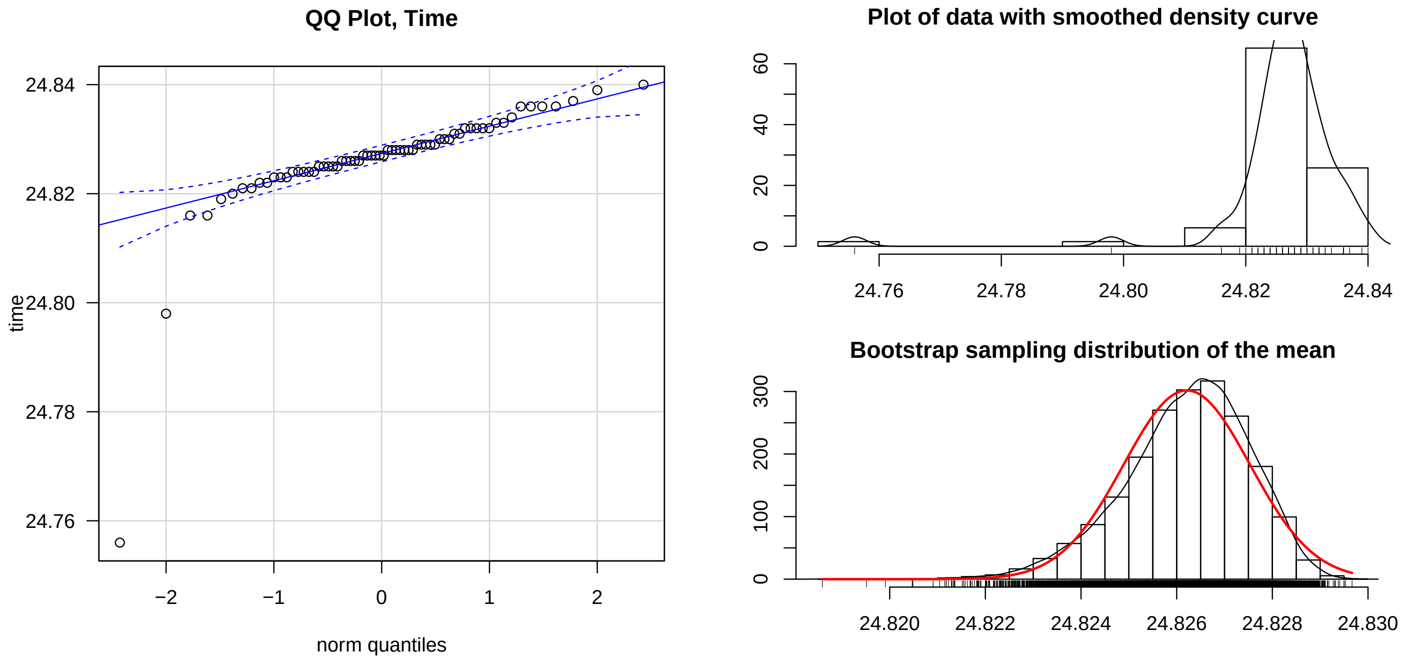
```
#### Example: Newcombe's Data
time <- c(24.828, 24.833, 24.834, 24.826, 24.824, 24.756
        , 24.827, 24.840, 24.829, 24.816, 24.798, 24.822
        , 24.824, 24.825, 24.823, 24.821, 24.830, 24.829
        , 24.831, 24.824, 24.836, 24.819, 24.820, 24.832
        , 24.836, 24.825, 24.828, 24.828, 24.821, 24.829
        , 24.837, 24.828, 24.830, 24.825, 24.826, 24.832
        , 24.836, 24.830, 24.836, 24.826, 24.822, 24.823
        , 24.827, 24.828, 24.831, 24.827, 24.827, 24.827
        , 24.826, 24.826, 24.832, 24.833, 24.832, 24.824
        , 24.839, 24.824, 24.832, 24.828, 24.825, 24.825
        , 24.829, 24.828, 24.816, 24.827, 24.829, 24.823)
library(nortest)
ad.test(time)

##
##  Anderson-Darling normality test
##
## data:  time
## A = 5.8843, p-value = 1.217e-14
```

```
# Histogram overlaid with kernel density curve
Passage_df <- data.frame(time)
p1 <- ggplot(Passage_df, aes(x = time))
  # Histogram with density instead of count on y-axis
```

```
p1 <- p1 + geom_histogram(aes(y=..density..), binwidth=0.001)
p1 <- p1 + geom_density(alpha=0.1, fill="white")
p1 <- p1 + geom_rug()

# violin plot
p2 <- ggplot(Passage_df, aes(x = "t", y = time))
p2 <- p2 + geom_violin(fill = "gray50")
p2 <- p2 + geom_boxplot(width = 0.2, alpha = 3/4)
p2 <- p2 + coord_flip()

# boxplot
p3 <- ggplot(Passage_df, aes(x = "t", y = time))
p3 <- p3 + geom_boxplot()
p3 <- p3 + coord_flip()

library(gridExtra)
grid.arrange(grobs = list(p1, p2, p3), ncol=1)
```



```
par(mfrow=c(1,1))
library(car)
qqPlot(time, las = 1, id = list(n = 0, cex = 1), lwd = 1, main="QQ Plot, Time")

bs.one.samp.dist(time)
```



The data set is skewed to the left, due to the presence of two extreme outliers that could potentially be misrecorded observations. Without additional information I would be hesitant to apply normal theory methods (the t-test), even though the sample size is "large" (bootstrap sampling distribution is still left-skewed). Furthermore, the t-test still suffers from a lack of robustness of sensitivity, even in large samples. A formal QQ-plot and normal test rejects, at the 0.01 level, the normality assumption needed for the standard methods.

The table below gives 95% t, sign, and Wilcoxon CIs. I am more comfortable with the sign CI for the population median than the Wilcoxon method, which assumes symmetry.

```
t.sum <- t.test(time)
t.sum$conf.int

## [1] 24.82357 24.82885
## attr(,"conf.level")
## [1] 0.95

diff(t.test(time)$conf.int)

## [1] 0.005283061

s.sum <- SIGN.test(time)
s.sum$conf.int

## [1] 24.82600 24.82849
## attr(,"conf.level")
## [1] 0.95

diff(s.sum$conf.int)
```

```
## [1] 0.00249297

w.sum <- wilcox.test(time, conf.int=TRUE)
w.sum$conf.int

## [1] 24.82604 24.82853
## attr(,"conf.level")
## [1] 0.95

diff(w.sum$conf.int)

## [1] 0.002487969
```

parameter	Method	CI Limits	Width
mean	t	(24.8236, 24.8289)	0.0053
median	sign	(24.8260, 24.8285)	0.0025
median	Wilcoxon	(24.8260, 24.8285)	0.0025

Note the big difference between the nonparametric and the t-CI. The nonparametric CIs are about 1/2 as wide as the t-CI. This reflects the impact that outliers have on the standard deviation, which directly influences the CI width.

6.5 Alternatives for ANOVA and Planned Comparisons

The classical ANOVA assumes that the populations have normal frequency curves and the populations have equal variances (or spreads). You learned formal tests for these assumptions in Chapter 5. When the assumptions do not hold, you can try one of the following two approaches. Before describing alternative methods, I will note that deviations from normality in one or more samples might be expected in a comparison involving many samples. You should downplay small deviations from normality in problems involving many samples.

6.5.1 Kruskal-Wallis ANOVA

The **Kruskal-Wallis** (KW) test is a non-parametric method for testing the hypothesis of equal population medians against the alternative that not all population medians are equal. The procedure assumes you have independent random samples from populations with frequency curves having **identical shapes and spreads**. The KW ANOVA is essentially the standard ANOVA based on ranked data. That is, we combine the samples, rank the observations from smallest to largest, and then return the ranks to the original samples and do the standard ANOVA using the ranks. The KW ANOVA is a multiple sample analog of the Wilcoxon-Mann-Whitney two sample procedure. Hence, multiple comparisons for a KW analysis, be they FSD or Bonferroni comparisons, are based on the two sample WMW procedure.

6.5.2 Transforming Data

The distributions in many data sets are skewed to the right with outliers. If the sample spreads, say s and IQR, increase with an increasing mean or median, you can often **transform data** to a scale where the normality and the constant spread assumption are more nearly satisfied. The transformed data are analyzed using the standard ANOVA. The two most commonly used transforms for this problem are the square root and natural logarithm, provided the data are nonnegative[4].

If the original distributions are nearly symmetric, but heavy-tailed, nonlinear transformations will tend to destroy the symmetry. Many statisticians recommend methods based on trimmed means for such data. These methods are not commonly used by other researchers.

[4] The aim behind the choice of a **variance-stabilizing transformation** is to find a simple function f to apply to values y in a data set to create new values $y' = f(y)$ such that the variability of the values y' is not related to their mean value. For example, suppose that the values y are realizations from a Poisson distribution. Because for the Poisson distribution the variance is identical to the mean, the variance varies with the mean. However, if the simple variance-stabilizing transformation $y' = \sqrt{y}$ is applied, the sampling variance will be independent of the mean. A few distributional examples are provided in the table below.

Distribution	Variance=g(mean)	Transformation $y' = f(y)$
Poisson	$\sigma^2 = \mu$	$y' = \sqrt{y}$
binomial	$\sigma^2 = \mu(1-\mu)$	$y' = \arcsin(\sqrt{(y)})$
lognormal	$\sigma^2 = \mu^2$	$y' = \log(y)$

Example: Hydrocarbon (HC) Emissions Data These data are the HC emissions at idling speed, in ppm, for automobiles of different years of manufacture. The data are a random sample of all automobiles tested at an Albuquerque shopping center. (It looks like we need to find some newer cars!)

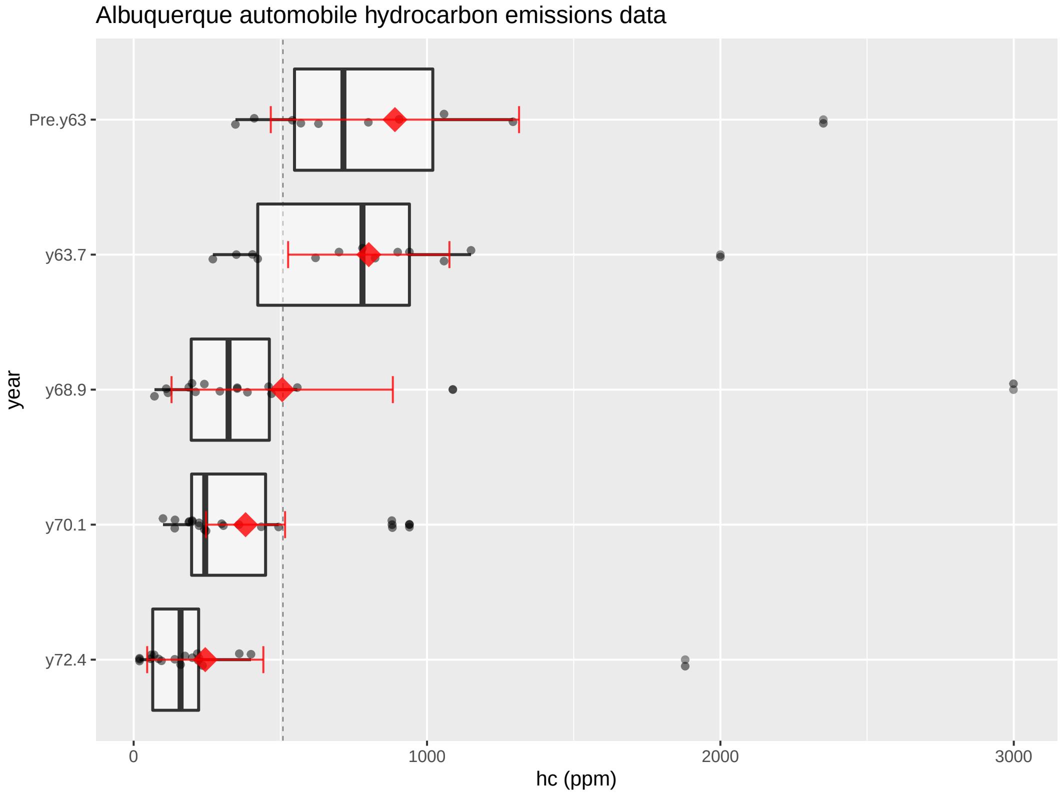
```
#### Example: Hydrocarbon (HC) Emissions Data
emis <- read.table(text="
 Pre-y63 y63-7 y68-9 y70-1 y72-4
    2351   620  1088   141   140
    1293   940   388   359   160
     541   350   111   247    20
    1058   700   558   940    20
     411  1150   294   882   223
     570  2000   211   494    60
     800   823   460   306    20
     630  1058   470   200    95
     905   423   353   100   360
     347   900    71   300    70
      NA   405   241   223   220
      NA   780  2999   190   400
      NA   270   199   140   217
      NA    NA   188   880    58
      NA    NA   353   200   235
      NA    NA   117   223  1880
      NA    NA    NA   188   200
      NA    NA    NA   435   175
      NA    NA    NA   940    85
      NA    NA    NA   241    NA
", header=TRUE)
#emis

# convert to long format
emis.long <- melt(emis,
              variable.name = "year",
              value.name = "hc",
              na.rm = TRUE
            )

## No id variables; using all as measure variables
# naming variables manually, the variable.name and value.name not working 11/2012
names(emis.long) <- c("year", "hc")
# summary of each year
by(emis.long$hc, emis.long$year, summary)

## emis.long$year: Pre.y63
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   347.0   548.2   715.0   890.6  1019.8  2351.0
## ----------------------------------------------------
## emis.long$year: y63.7
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   270.0   423.0   780.0   801.5   940.0  2000.0
## ----------------------------------------------------
## emis.long$year: y68.9
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    71.0   196.2   323.5   506.3   462.5  2999.0
## ----------------------------------------------------
## emis.long$year: y70.1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   100.0   197.5   244.0   381.4   449.8   940.0
## ----------------------------------------------------
## emis.long$year: y72.4
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    20.0    65.0   160.0   244.1   221.5  1880.0
```

```
# IQR and sd of each year
by(emis.long$hc, emis.long$year, function(X) { c(IQR(X), sd(X), length(X)) })
```

```
## emis.long$year: Pre.y63
## [1] 471.5000 591.5673  10.0000
## ----------------------------------------------------
## emis.long$year: y63.7
## [1] 517.0000 454.9285  13.0000
## ----------------------------------------------------
## emis.long$year: y68.9
## [1] 266.2500 707.8026  16.0000
## ----------------------------------------------------
## emis.long$year: y70.1
## [1] 252.2500 287.8864  20.0000
## ----------------------------------------------------
## emis.long$year: y72.4
## [1] 156.5000 410.7866  19.0000
```

```
# Plot the data using ggplot
library(ggplot2)
p <- ggplot(emis.long, aes(x = year, y = hc))
# plot a reference line for the global mean (assuming no groups)
p <- p + geom_hline(yintercept = mean(emis.long$hc),
                    colour = "black", linetype = "dashed", size = 0.3, alpha = 0.5)
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0), alpha = 0.5)
# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
                      colour = "red", alpha = 0.8)
# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
                      width = .2, colour = "red", alpha = 0.8)
p <- p + labs(title = "Albuquerque automobile hydrocarbon emissions data") + ylab("hc (ppm)")
```

```
# to reverse order that years print, so oldest is first on top
p <- p + scale_x_discrete(limits = rev(levels(emis.long$year)) )
p <- p + coord_flip()
p <- p + theme(legend.position="none")
print(p)
```



The standard ANOVA shows significant differences among the mean HC emissions. However, the standard ANOVA is inappropriate because the distributions are extremely skewed to the right due to presence of outliers in each sample.

```
fit.e <- aov(hc ~ year, data = emis.long)
summary(fit.e)
```

```
##             Df   Sum Sq Mean Sq F value  Pr(>F)
## year         4  4226834 1056709   4.343 0.00331 **
## Residuals   73 17759968  243287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit.e
```

```
## Call:
##    aov(formula = hc ~ year, data = emis.long)
##
## Terms:
##                     year Residuals
## Sum of Squares   4226834  17759968
## Deg. of Freedom        4        73
##
## Residual standard error: 493.2416
## Estimated effects may be unbalanced
```

The boxplots show that the typical HC emissions appear to decrease as the age of car increases (the simplest description). Although the spread in the samples, as measured by the IQR, also decreases as age increases, I am more comfortable with the KW ANOVA, in part because the KW analysis is not too sensitive to differences in spreads among samples. This point is elaborated upon later. As described earlier, the KW ANOVA is essentially an ANOVA based on the ranks. I give below the ANOVA based on ranks and the output from the KW procedure. They give similar p-values, and lead to the conclusion that there are significant differences among the population median HC emissions. A simple description is that the population median emission tends to decrease with the age of the car. You should follow up this analysis with Mann-Whitney multiple comparisons.

```
# ANOVA of rank, for illustration that this is similar to what KW is doing
fit.er <- aov(rank(hc) ~ year, data = emis.long)
summary(fit.er)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## year         4  16329    4082   12.85 5.74e-08 ***
## Residuals   73  23200     318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fit.er

## Call:
##    aov(formula = rank(hc) ~ year, data = emis.long)
##
## Terms:
##                     year Residuals
## Sum of Squares  16329.32  23199.68
## Deg. of Freedom        4        73
##
## Residual standard error: 17.82705
## Estimated effects may be unbalanced

# KW ANOVA
fit.ek <- kruskal.test(hc ~ year, data = emis.long)
fit.ek

##
```

```
##  Kruskal-Wallis rank sum test
##
## data:  hc by year
## Kruskal-Wallis chi-squared = 31.808, df = 4, p-value =
## 2.093e-06
```

It is common to transform the data to a log scale when the spread increases as the median or mean increases.

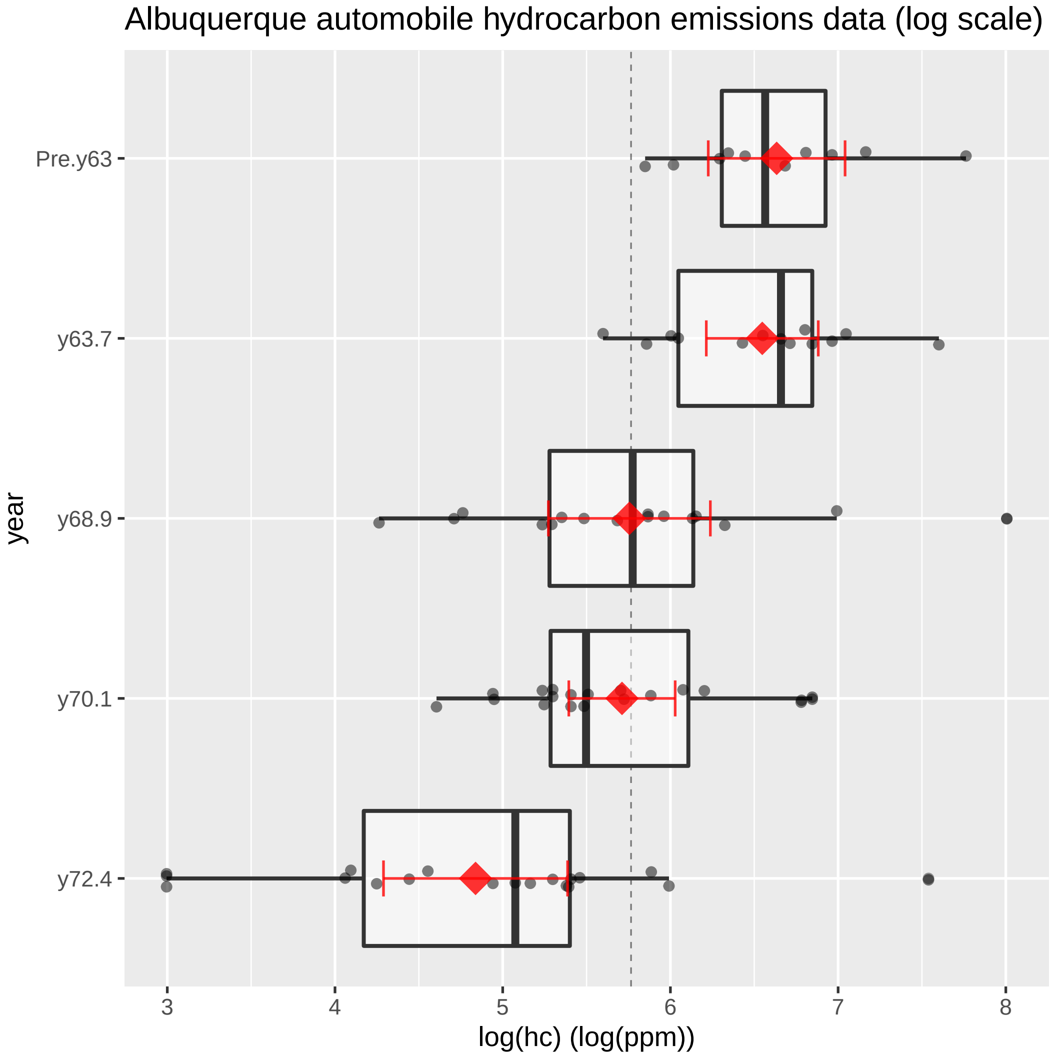
```
# log scale
emis.long$loghc <- log(emis.long$hc)
# summary of each year
by(emis.long$loghc, emis.long$year, summary)
```

```
## emis.long$year: Pre.y63
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.849   6.306   6.565   6.634   6.925   7.763
## ----------------------------------------------------
## emis.long$year: y63.7
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.598   6.047   6.659   6.548   6.846   7.601
## ----------------------------------------------------
## emis.long$year: y68.9
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.263   5.279   5.775   5.755   6.137   8.006
## ----------------------------------------------------
## emis.long$year: y70.1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.605   5.285   5.497   5.711   6.107   6.846
## ----------------------------------------------------
## emis.long$year: y72.4
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.996   4.171   5.075   4.838   5.400   7.539
```

```
# IQR and sd of each year
by(emis.long$loghc, emis.long$year, function(X) { c(IQR(X), sd(X), length(X)) })
```

```
## emis.long$year: Pre.y63
## [1]  0.6186119  0.5702081 10.0000000
## ----------------------------------------------------
## emis.long$year: y63.7
## [1]  0.7985077  0.5524878 13.0000000
## ----------------------------------------------------
## emis.long$year: y68.9
## [1]  0.8575139  0.9061709 16.0000000
## ----------------------------------------------------
## emis.long$year: y70.1
## [1]  0.8216494  0.6775933 20.0000000
## ----------------------------------------------------
## emis.long$year: y72.4
## [1]  1.228980  1.138882 19.000000
```

```
# Plot the data using ggplot
library(ggplot2)
p <- ggplot(emis.long, aes(x = year, y = loghc))
# plot a reference line for the global mean (assuming no groups)
p <- p + geom_hline(yintercept = mean(emis.long$loghc),
                    colour = "black", linetype = "dashed", size = 0.3, alpha = 0.5)
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0), alpha = 0.5)
# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
                      colour = "red", alpha = 0.8)
# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
                      width = .2, colour = "red", alpha = 0.8)
p <- p + labs(title = "Albuquerque automobile hydrocarbon emissions data (log scale)")
p <- p + ylab("log(hc) (log(ppm))")
# to reverse order that years print, so oldest is first on top
p <- p + scale_x_discrete(limits = rev(levels(emis.long$year)) )
p <- p + coord_flip()
p <- p + theme(legend.position="none")
print(p)
```



After transformation, the samples have roughly the same spread (IQR and s) and shape. The transformation does not completely eliminate the outliers. However, I am more comfortable with a standard ANOVA on this scale than with the original data. A difficulty here is that the ANOVA is comparing population mean log HC emission (so interpretations are on the log ppm scale, instead of the natural ppm scale). Summaries for the ANOVA on the log hydrocarbon emissions levels are given below.

```
# ANOVA of rank, for illustration that this is similar to what KW is doing
fit.le <- aov(loghc ~ year, data = emis.long)
summary(fit.le)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## year         4  31.90   7.974   11.42 2.98e-07 ***
## Residuals   73  50.98   0.698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit.le

## Call:
##    aov(formula = loghc ~ year, data = emis.long)
##
## Terms:
##                      year Residuals
## Sum of Squares  31.89510  50.97679
## Deg. of Freedom        4        73
##
## Residual standard error: 0.8356508
## Estimated effects may be unbalanced

# KW ANOVA -- same conclusions as original scale, since based on ranks
fit.lek <- kruskal.test(loghc ~ year, data = emis.long)
fit.lek

##
##  Kruskal-Wallis rank sum test
##
## data:  loghc by year
## Kruskal-Wallis chi-squared = 31.808, df = 4, p-value =
## 2.093e-06
```

The boxplot of the log-transformed data reinforces the reasonableness of the original KW analysis. Why? The log-transformed distributions have fairly similar shapes and spreads, so a KW analysis on these data is sensible. The ranks for the original and log-transformed data are identical, so the KW analyses on the log-transformed data and the original data must lead to the same conclusions. This suggests that the KW ANOVA is not overly sensitive to differences in spreads among the samples.

There are two reasonable analyses here: the standard ANOVA using log HC emissions, and the KW analysis of the original data. The first analysis gives a comparison of mean log HC emissions. The second involves a comparison of median HC emissions. A statistician would present both analyses to the scientist who collected the data to make a decision on which was more meaningful (independently of the results[5]!). Multiple comparisons would be performed relative to the selected analysis (t-tests for ANOVA or WMW-tests for KW ANOVA).

[5] It is unethical to choose a method based on the results it gives.

Example: Hodgkin's Disease Study Plasma bradykininogen levels were measured in normal subjects, in patients with active Hodgkin's disease, and in patients with inactive Hodgkin's disease. The globulin bradykininogen is the precursor substance for bradykinin, which is thought to be a chemical mediator of inflammation. The data (in micrograms of bradykininogen per milliliter of plasma) are displayed below. The three samples are denoted by **nc** for normal controls, **ahd** for active Hodgkin's disease patients, and **ihd** for inactive Hodgkin's disease patients.

The medical investigators wanted to know if the three samples differed in their bradykininogen levels. Carry out the statistical analysis you consider to be most appropriate, and state your conclusions to this question.

Read in the data, look at summaries on the original scale, and create a plot. Also, look at summaries on the log scale and create a plot.

```
#### Example: Hodgkin's Disease Study
hd <- read.table(text="
    nc   ahd    ihd
  5.37  3.96   5.37
  5.80  3.04  10.60
  4.70  5.28   5.02
  5.70  3.40  14.30
  3.40  4.10   9.90
  8.60  3.61   4.27
  7.48  6.16   5.75
  5.77  3.22   5.03
  7.15  7.48   5.74
  6.49  3.87   7.85
  4.09  4.27   6.82
  5.94  4.05   7.90
  6.38  2.40   8.36
  9.24  5.81   5.72
  5.66  4.29   6.00
  4.53  2.77   4.75
  6.51  4.40   5.83
  7.00    NA   7.30
  6.20    NA   7.52
  7.04    NA   5.32
  4.82    NA   6.05
  6.73    NA   5.68
  5.26    NA   7.57
    NA    NA   5.68
    NA    NA   8.91
    NA    NA   5.39
    NA    NA   4.40
    NA    NA   7.13
", header=TRUE)
#hd

# convert to long format
hd.long <- melt(hd,
              variable.name = "patient",
              value.name = "level",
              na.rm = TRUE
            )
## No id variables; using all as measure variables
# naming variables manually, the variable.name and value.name not working 11/2012
names(hd.long) <- c("patient", "level")
# summary of each patient
by(hd.long$level, hd.long$patient, summary)
```

```
# IQR and sd of each patient
by(hd.long$level, hd.long$patient, function(X) { c(IQR(X), sd(X), length(X)) })
```

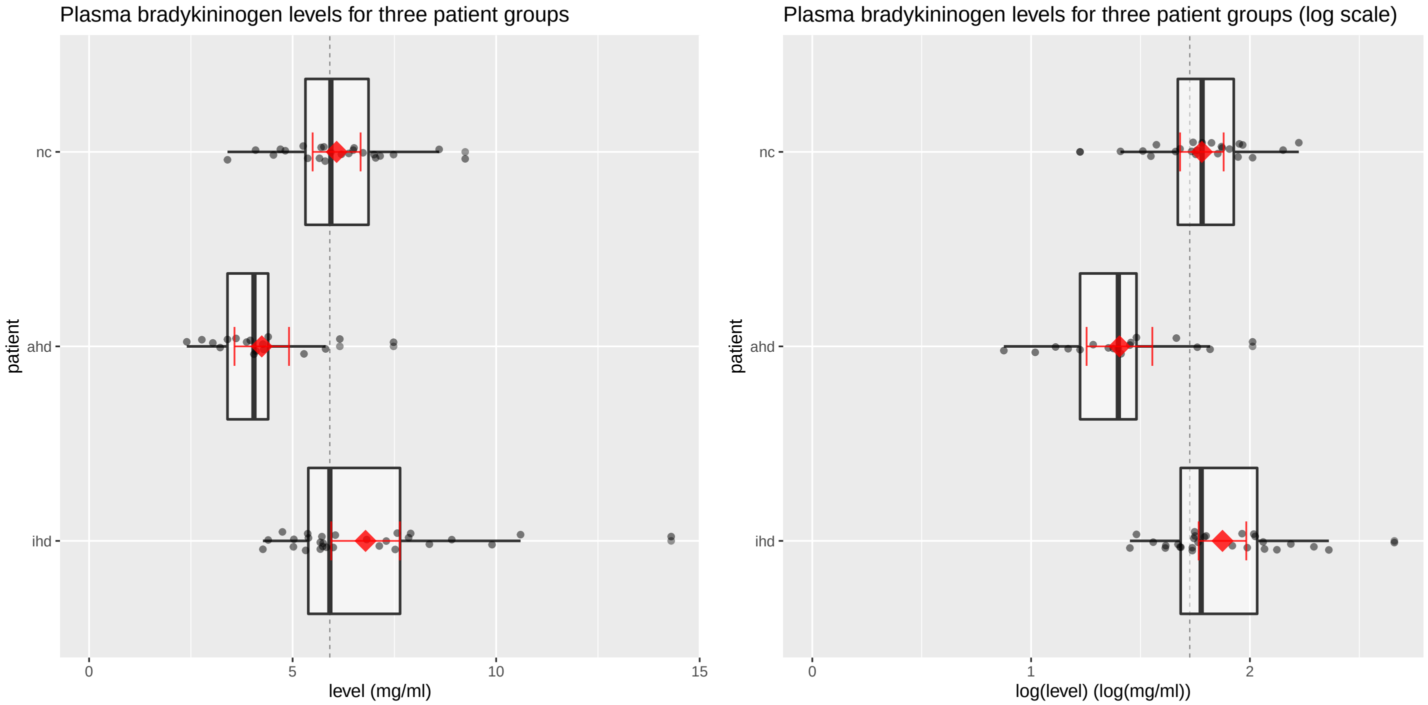
```
# log scale
hd.long$loglevel <- log(hd.long$level)
# summary of each patient
by(hd.long$loglevel, hd.long$patient, summary)
## hd.long$patient: nc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.224   1.670   1.782   1.780   1.926   2.224
## ----------------------------------------------------
## hd.long$patient: ahd
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8755  1.2238  1.3987  1.4039  1.4816  2.0122
## ----------------------------------------------------
## hd.long$patient: ihd
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.452   1.684   1.777   1.875   2.033   2.660
```

```
# IQR and sd of each patient
by(hd.long$loglevel, hd.long$patient, function(X) { c(IQR(X), sd(X), length(X)) })
```

```
## hd.long$patient: nc
## [1]  0.2557632  0.2303249 23.0000000
## ----------------------------------------------------
## hd.long$patient: ahd
## [1]  0.2578291  0.2920705 17.0000000
## ----------------------------------------------------
## hd.long$patient: ihd
## [1]  0.3496572  0.2802656 28.0000000
```

```
# Plot the data using ggplot
library(ggplot2)
p <- ggplot(hd.long, aes(x = patient, y = level))
# plot a reference line for the global mean (assuming no groups)
p <- p + geom_hline(yintercept = mean(hd.long$level),
                    colour = "black", linetype = "dashed", size = 0.3, alpha = 0.5)
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0), alpha = 0.5)
# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
                      colour = "red", alpha = 0.8)
# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
                      width = .2, colour = "red", alpha = 0.8)
p <- p + labs(title = "Plasma bradykininogen levels for three patient groups")
p <- p + ylab("level (mg/ml)")
# to reverse order that years print, so oldest is first on top
p <- p + scale_x_discrete(limits = rev(levels(hd.long$patient)) )
p <- p + ylim(c(0,max(hd.long$level)))
p <- p + coord_flip()
p <- p + theme(legend.position="none")
print(p)

## log scale
# Plot the data using ggplot
library(ggplot2)
p <- ggplot(hd.long, aes(x = patient, y = loglevel))
# plot a reference line for the global mean (assuming no groups)
p <- p + geom_hline(yintercept = mean(hd.long$loglevel),
                    colour = "black", linetype = "dashed", size = 0.3, alpha = 0.5)
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(position = position_jitter(w = 0.05, h = 0), alpha = 0.5)
# diamond at mean for each group
```

```
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
                      colour = "red", alpha = 0.8)
# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
                      width = .2, colour = "red", alpha = 0.8)
p <- p + labs(title = "Plasma bradykininogen levels for three patient groups (log scale)")
p <- p + ylab("log(level) (log(mg/ml))")
# to reverse order that years print, so oldest is first on top
p <- p + scale_x_discrete(limits = rev(levels(hd.long$patient)) )
p <- p + ylim(c(0,max(hd.long$loglevel)))
p <- p + coord_flip()
p <- p + theme(legend.position="none")
print(p)
```



Although the spread (IQR, s) in the *ihd* sample is somewhat greater than the spread in the other samples, the presence of skewness and outliers in the boxplots is a greater concern regarding the use of the classical ANOVA. The shapes and spreads in the three samples are roughly identical, so a Kruskal-Wallis nonparametric ANOVA appears ideal. As a sidelight, I transformed plasma levels to a log scale to reduce the skewness and eliminate the outliers. The boxplots of the transformed data show reasonable symmetry across groups, but outliers are still present. I will stick with the Kruskal-Wallis ANOVA (although it would not be much of a problem to use the classical ANOVA on transformed data).

Let η_{nc} = population median plasma level for normal controls, η_{ahd} = population median plasma level for active Hodgkin's disease patients, and η_{ihd} = population median plasma level for inactive Hodgkin's disease patients. The KW test of $H_0 : \eta_{nc} = \eta_{ahd} = \eta_{ihd}$ versus H_A : not H_0 is highly significant (p-value= 0.00003), suggesting differences among the population median plasma levels. The Kruskal-Wallis ANOVA summary is given below.

```
# KW ANOVA
fit.h <- kruskal.test(level ~ patient, data = hd.long)
fit.h

##
##  Kruskal-Wallis rank sum test
##
## data:  level by patient
## Kruskal-Wallis chi-squared = 20.566, df = 2, p-value =
## 3.421e-05
```

I followed up the KW ANOVA with Bonferroni comparisons of the samples, using the Mann-Whitney two sample procedure. There are three comparisons, so an overall FER of 0.05 is achieved by doing the individual tests at the 0.05/3=0.0167 level. Alternatively, you can use 98.33% CI for differences in population medians.

```
# with continuity correction in the normal approximation for the p-value
wilcox.test(hd$nc , hd$ahd, conf.int=TRUE, conf.level = 0.9833)

## Warning in wilcox.test.default(hd$nc, hd$ahd, conf.int = TRUE, conf.level = 0.9833):  canno
compute exact p-value with ties
## Warning in wilcox.test.default(hd$nc, hd$ahd, conf.int = TRUE, conf.level = 0.9833):  canno
compute exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hd$nc and hd$ahd
## W = 329, p-value = 0.0002735
## alternative hypothesis: true location shift is not equal to 0
## 98.33 percent confidence interval:
##  0.8599458 2.9000789
## sample estimates:
## difference in location
##               1.910067

wilcox.test(hd$nc , hd$ihd, conf.int=TRUE, conf.level = 0.9833)

## Warning in wilcox.test.default(hd$nc, hd$ihd, conf.int = TRUE, conf.level = 0.9833):  canno
compute exact p-value with ties
```

```
## Warning in wilcox.test.default(hd$nc, hd$ihd, conf.int = TRUE, conf.level = 0.9833):  canno
compute exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hd$nc and hd$ihd
## W = 276.5, p-value = 0.3943
## alternative hypothesis: true location shift is not equal to 0
## 98.33 percent confidence interval:
##  -1.5600478  0.6800262
## sample estimates:
## difference in location
##             -0.3413932
```

```
wilcox.test(hd$ahd, hd$ihd, conf.int=TRUE, conf.level = 0.9833)
```

```
## Warning in wilcox.test.default(hd$ahd, hd$ihd, conf.int = TRUE, conf.level = 0.9833):
cannot compute exact p-value with ties
## Warning in wilcox.test.default(hd$ahd, hd$ihd, conf.int = TRUE, conf.level = 0.9833):
cannot compute exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hd$ahd and hd$ihd
## W = 56, p-value = 2.143e-05
## alternative hypothesis: true location shift is not equal to 0
## 98.33 percent confidence interval:
##  -3.500059 -1.319957
## sample estimates:
## difference in location
##              -2.146666
```

The only comparison with a p-value greater than 0.0167 involved the **nc** and **ihd** samples. The comparison leads to two groups, and is consistent with what we see in the boxplots.

```
ahd   nc   ihd
---   --------
```

You have sufficient evidence to conclude that the plasma bradykininogen levels for active Hodgkin's disease patients (ahd) is lower than the population median levels for normal controls (nc) and for patients with inactive Hodgkin's disease (ihd). You do not have sufficient evidence to conclude that the population median levels for normal controls (nc) and for patients with inactive Hodgkin's disease (ihd) are different. The CIs give an indication of size of differences in the population medians.

6.5.3 Planned Comparisons

Bonferroni multiple comparisons are generally preferred to Fisher's least significant difference approach. Fisher's method does not control the familywise error rate and produces too many spurious significant differences (claims of significant differences that are due solely to chance variation and not to actual differences in population means). However, Bonferroni's method is usually very conservative when a large number of comparisons is performed — large differences in sample means are needed to claim significance. A way to reduce this conservatism is to avoid doing all possible comparisons. Instead, one should, when possible, decide *a priori* (before looking at the data) which comparisons are of primary interest, and then perform only those comparisons.

For example, suppose a medical study compares five new treatments with a control (a six group problem). The medical investigator may not be interested in all 15 possible comparisons, but only in which of the five treatments differ on average from the control. Rather than performing the 15 comparisons, each at the say $0.05/15 = 0.0033$ level, she could examine the five comparisons of interest at the $0.05/5 = 0.01$ level. By deciding beforehand which comparisons are of interest, she can justify using a 0.01 level for the comparisons, instead of the more conservative 0.0033 level needed when doing all possible comparisons.

To illustrate this idea, consider the KW analysis of HC emissions. We saw that there are significant differences among the population median HC emissions. Given that the samples have a natural ordering

Sample	Year of manufacture
1	Pre-1963
2	63 – 67
3	68 – 69
4	70 – 71
5	72 – 74

you may primarily be interested in whether the population medians for cars manufactured in consecutive samples are identical. That is, you may be primarily interested in the following 4 comparisons:

Pre-1963	vs	63 – 67
63 – 67	vs	68 – 69
68 – 69	vs	70 – 71
70 – 71	vs	72 – 74

A Bonferroni analysis would carry out each comparison at the $0.05/4 = 0.0125$ level versus the $0.05/10 = 0.005$ level when all comparisons are done.

The following output was obtained for doing these four comparisons, based on Wilcoxon-Mann-Whitney two-sample tests (why?[6]). Two-year groups are claimed to be different if the p-value is 0.0125 or below, or equivalently, if a 98.75% CI for the difference in population medians does not contain zero.

```
#### Planned Comparisons
# with continuity correction in the normal approximation for the p-value
wilcox.test(emis$y63.7, emis$Pre.y63, conf.int=TRUE, conf.level = 0.9875)

## Warning in wilcox.test.default(emis$y63.7, emis$Pre.y63, conf.int = TRUE, :  cannot compute
exact p-value with ties
## Warning in wilcox.test.default(emis$y63.7, emis$Pre.y63, conf.int = TRUE, :  cannot compute
exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  emis$y63.7 and emis$Pre.y63
## W = 61.5, p-value = 0.8524
## alternative hypothesis: true location shift is not equal to 0
## 98.75 percent confidence interval:
##  -530.0001  428.0000
## sample estimates:
## difference in location
##               -15.4763

wilcox.test(emis$y68.9, emis$y63.7  , conf.int=TRUE, conf.level = 0.9875)

## Warning in wilcox.test.default(emis$y68.9, emis$y63.7, conf.int = TRUE, :  cannot compute
exact p-value with ties
## Warning in wilcox.test.default(emis$y68.9, emis$y63.7, conf.int = TRUE, :  cannot compute
exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  emis$y68.9 and emis$y63.7
## W = 43, p-value = 0.007968
```

[6] The ANOVA is the multi-sample analog to the two-sample t-test for the mean, and the KW ANOVA is the multi-sample analog to the WMW two-sample test for the median. Thus, we follow up a KW ANOVA with WMW two-sample tests at the chosen multiple comparison adjusted error rate.

```
## alternative hypothesis: true location shift is not equal to 0
## 98.75 percent confidence interval:
##  -708.99999  -51.99998
## sample estimates:
## difference in location
##              -397.4227
```

```
wilcox.test(emis$y70.1, emis$y68.9  , conf.int=TRUE, conf.level = 0.9875)
```

```
## Warning in wilcox.test.default(emis$y70.1, emis$y68.9, conf.int = TRUE, :  cannot compute
exact p-value with ties
## Warning in wilcox.test.default(emis$y70.1, emis$y68.9, conf.int = TRUE, :  cannot compute
exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  emis$y70.1 and emis$y68.9
## W = 156, p-value = 0.9112
## alternative hypothesis: true location shift is not equal to 0
## 98.75 percent confidence interval:
##  -206.0001  171.0000
## sample estimates:
## difference in location
##              -10.99997
```

```
wilcox.test(emis$y72.4, emis$y70.1  , conf.int=TRUE, conf.level = 0.9875)
```

```
## Warning in wilcox.test.default(emis$y72.4, emis$y70.1, conf.int = TRUE, :  cannot compute
exact p-value with ties
## Warning in wilcox.test.default(emis$y72.4, emis$y70.1, conf.int = TRUE, :  cannot compute
exact confidence intervals with ties
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  emis$y72.4 and emis$y70.1
## W = 92.5, p-value = 0.006384
## alternative hypothesis: true location shift is not equal to 0
## 98.75 percent confidence interval:
##  -285.999962   -6.000058
## sample estimates:
## difference in location
##                   -130
```

There are significant differences between the 1963-67 and 1968-69 samples, and between the 1970-71 and 1972-74 samples. You are 98.75% confident that the population median HC emissions for 1963-67 year cars is between 52 and 708.8 ppm greater than the population median for 1968-69 cars. Similarly, you are 98.75% confident that the population median HC emissions for 1970-71 year cars is between 6.1 and 285.9 ppm greater than the population median for 1972-

74 cars. Overall, you are 95% confident among the four pairwise comparisons that you have not declared a difference significant when it isn't.

6.5.4 Two final ANOVA comments

It is not uncommon for researchers to combine data from groups not found to be significantly different. This is not, in general, a good practice. Just because you do not have sufficient evidence to show differences does not imply that you should treat the groups as if they are the same!

If the data distributions do not substantially deviate from normality, but the spreads are different across samples, you might consider the standard ANOVA followed with multiple comparisons using two-sample tests based on Satterthwaite's approximation.

6.6 Permutation tests

Permutation tests[7] are a subset of non-parametric statistics. The basic premise is to use only the assumption that it is possible that all of the treatment groups are equivalent, and that every member of them is the same before sampling began (i.e., the position in the group to which they belong is not differentiable from other position before the positions are filled). From this, one can calculate a statistic and then see to what extent this statistic is special by seeing how likely it would be if the group assignments had been jumbled.

A permutation test (also called a randomization test, re-randomization test, or an exact test) is a type of statistical significance test in which the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under **rearrangements of the labels** on the observed data points. In other words, the method by which treatments are allocated to subjects in an experimental design is mirrored in the analysis of that design. If the labels are exchangeable under the null hypothesis, then the

[7] `http://en.wikipedia.org/wiki/Resampling_(statistics)`

resulting tests yield exact significance levels. Confidence intervals can then be derived from the tests. The theory has evolved from the works of R.A. Fisher and E.J.G. Pitman in the 1930s.

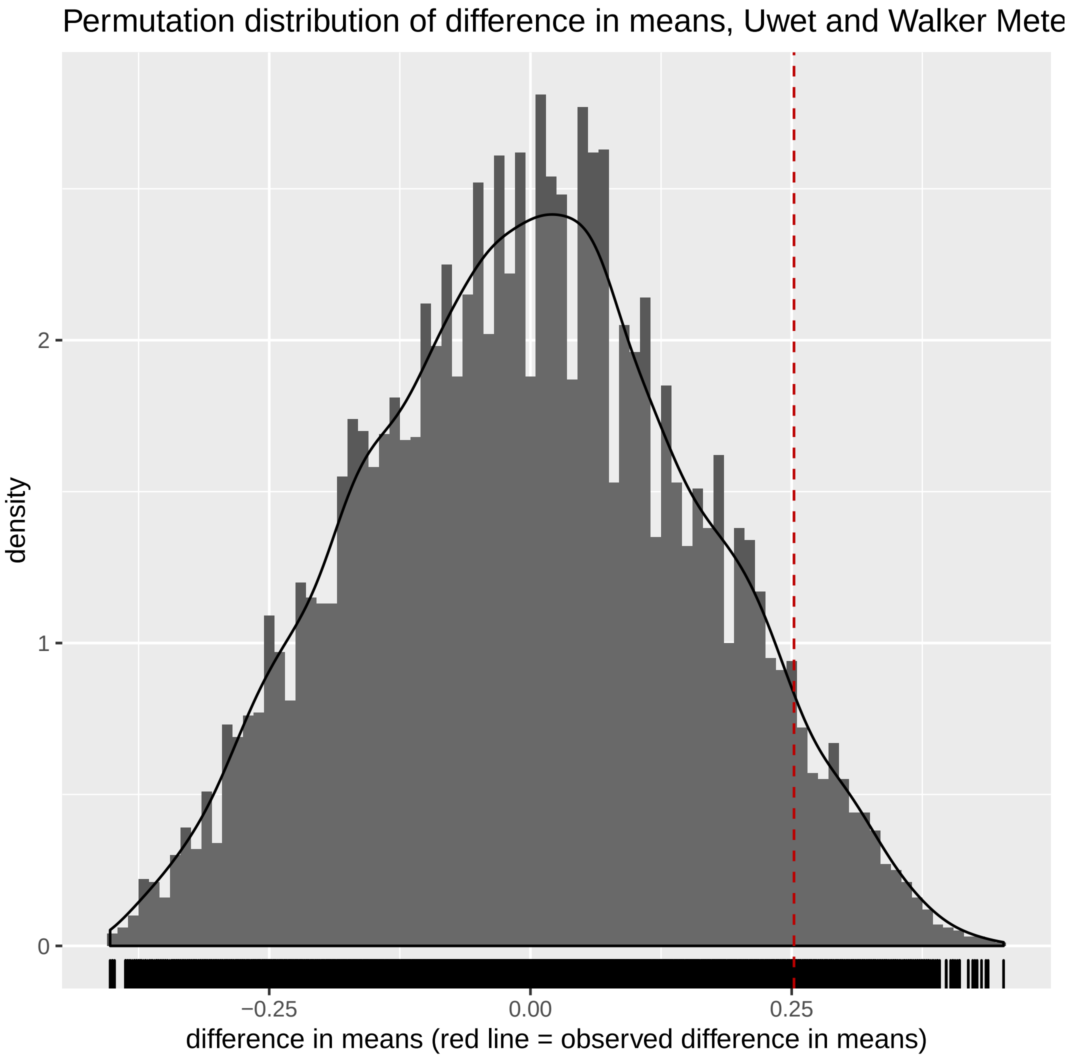
Let's illustrate the basic idea of a permutation test using the Meteorites example. Suppose we have two groups Uwet and Walker whose sample means are $\bar{Y}_{\text{U}}$ and $\bar{Y}_{\text{W}}$, and that we want to test, at 5% significance level, whether they come from the same distribution. Let $n_{\text{U}} = 9$ and $n_{\text{W}} = 10$ be the sample size corresponding to each group. The permutation test is designed to determine whether the observed difference between the sample means is large enough to reject the null hypothesis $H_0 : \mu_{\text{U}} = \mu_{\text{W}}$, that the two groups have identical means.

The test proceeds as follows. First, the difference in means between the two samples is calculated: this is the observed value of the test statistic, $T_{(\text{obs})}$. Then the observations of groups Uwet and Walker are pooled.

```
#### Permutation tests
# Calculated the observed difference in means
        # met.long includes both Uwet and Walker groups
Tobs <- mean(met.long[(met.long$site == "Uwet"  ), 2]) -
        mean(met.long[(met.long$site == "Walker"), 2])
Tobs

## [1] 0.2522222
```

Next, the difference in sample means is calculated and recorded for every possible way of dividing these pooled values into two groups of size $n_{\text{U}} = 9$ and $n_{\text{W}} = 10$ (i.e., for every permutation of the group labels Uwet and Walker). The set of these calculated differences is the exact distribution of possible differences under the null hypothesis that group label does not matter. This exact distribution can be approximated by drawing a large number of random permutations.

```
# Plan:
#   Initialize a vector in which to store the R number of difference of means.
#   Calculate R differences in means for R permutations, storing the results.
#     Note that there are prod(1:19) = 10^17 total permutations,
#       but the R repetitions will serve as a good approximation.
#   Plot the permutation null distribution with an indication of the Tobs.

# R = a large number of repetitions
R <- 1e4
# initialize the vector of difference of means from the permutations
Tperm <- rep(NA, R)
# For each of R repetitions, permute the Uwet and Walker labels,
#   calculate the difference of means with the permuted labels,
#   and store the result in the i.R'th position of Tperm.
for (i.R in 1:R) {
  # permutation of 19 = 9+10 integers 1, 2, ..., 19
  ind.perm <- sample.int(nrow(met.long))
  # identify as "TRUE" numbers 1, ..., 9 (the number of Uwet labels)
  lab.U <- (ind.perm <= sum(met.long$site == "Uwet"))                 #£
  # identify as "TRUE" numbers 10, ..., 19 (the number of Walker labels)
  #   that is, all the non-Uwet labels
  lab.W <- !lab.U

  # calculate the difference in means and store in Tperm at index i.R
  Tperm[i.R] <- mean(met.long[lab.U, 2]) - mean(met.long[lab.W, 2])
}

# Plot the permutation null distribution with an indication of the Tobs.
dat <- data.frame(Tperm)

library(ggplot2)
p <- ggplot(dat, aes(x = Tperm))
#p <- p + scale_x_continuous(limits=c(-20,+20))
p <- p + geom_histogram(aes(y=..density..), binwidth=0.01)
p <- p + geom_density(alpha=0.1, fill="white")
p <- p + geom_rug()
# vertical line at Tobs
p <- p + geom_vline(aes(xintercept=Tobs), colour="#BB0000", linetype="dashed")
p <- p + labs(title = "Permutation distribution of difference in means, Uwet and Walker Meteor
p <- p + xlab("difference in means (red line = observed difference in means)")
print(p)
```



Notice the contrast in this permutation distribution of the difference in means from a normal distribution.

The one-sided p-value of the test is calculated as the proportion of sampled permutations where the difference in means was at least as extreme as $T_{\mathrm{(obs)}}$. The two-sided p-value of the test is calculated as the proportion of sampled permutations where the absolute difference was at least as extreme as $|T_{\mathrm{(obs)}}|$.

```
# Calculate a two-sided p-value.
p.upper <- sum((Tperm >= abs(Tobs))) / R
p.upper

## [1] 0.0592

p.lower <- sum((Tperm <= -abs(Tobs))) / R
p.lower

## [1] 0.0599

p.twosided <- p.lower + p.upper
p.twosided
```

```
## [1] 0.1191
```

Note that the two-sided p-value of 0.1191 is consistent, in this case, with the two-sample t-test p-values of 0.1134 (pooled) and 0.1290 (Satterthwaite), but different from 0.0497 (WMW). The permutation is a comparison of means **without** the normality assumption, though requires that the observations are exchangable between populations under H_0.

If the only purpose of the test is reject or not reject the null hypothesis, we can as an alternative sort the recorded differences, and then observe if $T_{(\mathrm{obs})}$ is contained within the middle 95% of them. If it is not, we reject the hypothesis of equal means at the 5% significance level.

6.6.1 Linear model permutation tests in R

The `coin` package provides an implementation of a general framework for conditional inference procedures commonly known as permutation tests. In the help on `?"coin-package"` search for `location` to find tests for the means or medians of populations (such as `oneway_test()`). Other packages of note include `perm` and `exactRankTests` (`lmPerm` is defunct).

Below I calculate the standard t-test for the Meteorite data using `t.test()` and `lm()`, then compare that with `oneway_test()` and what we calculated using our calculation of the permutation test.

```
# standard two-sample t-test with equal variances
t.summary <- t.test(cool ~ site, data = met.long, var.equal = TRUE)
t.summary

##
##  Two Sample t-test
##
## data:  cool by site
## t = 1.6689, df = 17, p-value = 0.1134
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0666266  0.5710710
## sample estimates:
##   mean in group Uwet mean in group Walker
##            0.4522222            0.2000000

# linear model form of t-test, "siteWalker" has estimate, se, t-stat, and p-value
lm.summary <- lm(cool ~ site, data = met.long)
```

```
summary(lm.summary)

##
## Call:
## lm(formula = cool ~ site, data = met.long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2922 -0.1961 -0.1600  0.0250  0.7478
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4522     0.1096   4.125 0.000708 ***
## siteWalker   -0.2522     0.1511  -1.669 0.113438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3289 on 17 degrees of freedom
## Multiple R-squared:  0.1408,Adjusted R-squared:  0.09024
## F-statistic: 2.785 on 1 and 17 DF,  p-value: 0.1134

# permutation test version
library(coin)
# Fisher-Pitman permutation test
oneway.summary <- oneway_test(cool ~ site, data = met.long, conf.int = TRUE)
oneway.summary

##
##  Asymptotic Two-Sample Fisher-Pitman Permutation Test
##
## data:  cool by site (Uwet, Walker)
## Z = 1.5919, p-value = 0.1114
## alternative hypothesis: true mu is not equal to 0

  # examples of extracting values from coins S4 class objects
  coin::expectation(oneway.summary)

##     Uwet
## 2.875263

  coin::covariance(oneway.summary)

##           Uwet
## Uwet 0.5632881

  coin::pvalue(oneway.summary)

## [1] 0.1114144

  #coin::confint(oneway.summary)
```

The permutation test gives a p-value of 0.1114 which is close to our manually calculated permuatation p-value of 0.1191.

For the emisions data, below we compare the ANOVA results (assuming normality) with a permutation test without distributional assumptions.

```
fit.e <- aov(hc ~ year, data = emis.long)
summary(fit.e)
```

```
##             Df   Sum Sq Mean Sq F value  Pr(>F)
## year         4  4226834 1056709   4.343 0.00331 **
## Residuals   73 17759968  243287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(coin)
# Fisher-Pitman permutation test
oneway.summary <- oneway_test(hc ~ year, data = emis.long)
oneway.summary
```

```
##
##  Asymptotic K-Sample Fisher-Pitman Permutation Test
##
## data:  hc by
##   year (Pre.y63, y63.7, y68.9, y70.1, y72.4)
## chi-squared = 14.803, df = 4, p-value = 0.005128
```

Thus the permutation test of the ANOVA hypothesis on means rejects the null hypothesis of all equal means. A followup set of pairwise tests can be done by looping over pairs of factors.

First we list the factor levels ordered by their medians, the ordering by medians is helpful at the end when the results of the pairwise comparisons are given.

```
# these are the levels of the factor, ordered by their medians
fac.lev <- levels(reorder(levels(emis.long$year)
                        , -as.numeric(by(emis.long$loghc, emis.long$year, median)))
                )
fac.lev
```

```
## [1] "y63.7"   "Pre.y63" "y68.9"   "y70.1"   "y72.4"
```

Create a matrix to store pairwise comparison p-values, then loop over all pairs of groups and perform a two-sample permutation test. Store the p-value for each test in the matrix.

```
# create a matrix to store pairwise comparison p-values
mc.pval <- matrix(NA
                , nrow = length(fac.lev)
                , ncol = length(fac.lev)
                , dimnames = list(fac.lev, fac.lev))
# diag is always 1, no group differs from itself
diag(mc.pval) <- 1
mc.pval
```

```
##         y63.7 Pre.y63 y68.9 y70.1 y72.4
## y63.7       1      NA    NA    NA    NA
## Pre.y63    NA       1    NA    NA    NA
## y68.9      NA      NA     1    NA    NA
## y70.1      NA      NA    NA     1    NA
## y72.4      NA      NA    NA    NA     1
```

```
# loop over all pairs of factor levels, perform two-sample test,
#   and store p-value in matrix
for (i1 in 1:(length(fac.lev) - 1)) {
  for (i2 in (i1 + 1):length(fac.lev)) {
    ## DEBUG - to make sure the indexing is working, you can print them:
    # print(cat(i1, i2))

    library(coin)
    # Fisher-Pitman permutation test
    oneway.summary <- oneway_test(hc ~ year, data = subset(emis.long, (year == fac.lev[i1] | y

    # put p-value in matrix
    mc.pval[i1, i2] <- coin::pvalue(oneway.summary)
    mc.pval[i2, i1] <- mc.pval[i1, i2]
  }
}

# p-values
mc.pval
```

```
##               y63.7     Pre.y63     y68.9       y70.1       y72.4
## y63.7   1.000000000 0.676572596 0.1993877 0.004273746 0.002185513
## Pre.y63 0.676572596 1.000000000 0.1611790 0.005319987 0.003379156
## y68.9   0.199387725 0.161179041 1.0000000 0.468455149 0.177187250
## y70.1   0.004273746 0.005319987 0.4684551 1.000000000 0.227517382
## y72.4   0.002185513 0.003379156 0.1771873 0.227517382 1.000000000
```

Summarize the results of the pairwise comparisons. Groups with a common letter are not statistically different.

```
# summary of pairwise comparisons
#   threshold is Bonferroni-corrected alpha=0.05 / 10
library(multcompView)
multcompLetters( mc.pval
               , compare = "<"
               , threshold = 0.05 / choose(length(fac.lev), 2)
               , Letters = letters
               , reversed = FALSE)
```

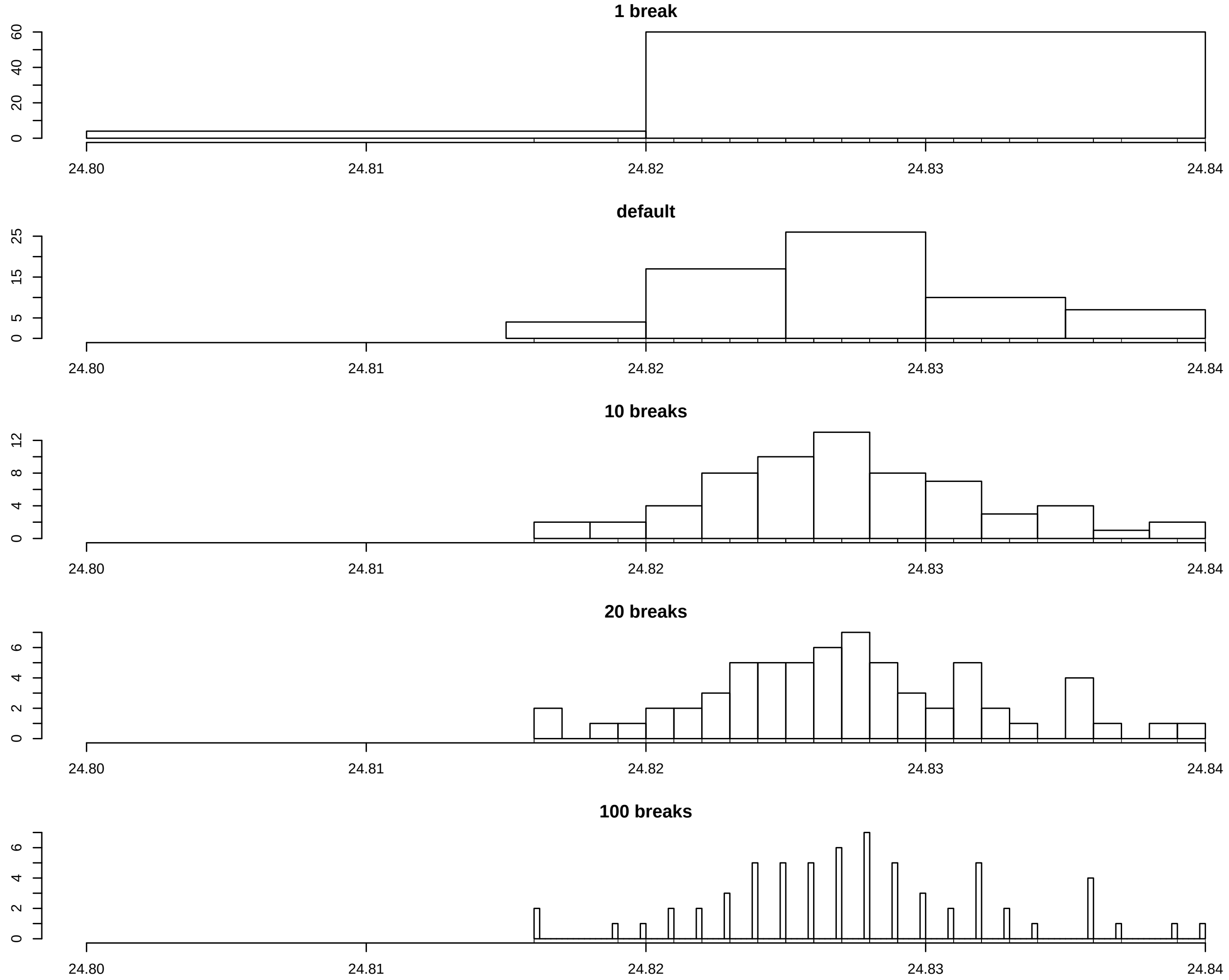
```
##   y63.7 Pre.y63   y68.9   y70.1   y72.4
##     "a"    "ab"   "abc"    "bc"     "c"
```

6.7 Density estimation

Density estimation is like a histogram: It is a method for visualizing the shape of a univariate distribution (there are methods for doing multivariate density estimation as well, but we will ignore those for the time being). In fact, I snuck in density estimation in the first chapter and have been using it all along! Let's experiment with Newcombe's speed-of-light data (excluding the two outliers).

Consider the shape of the histogram for different numbers of bins.

```
#### Density estimation
# include time ranks 3 and above, that is, remove the lowest two values
time2 <- time[(rank(time) >= 3)]

  old.par <- par(no.readonly = TRUE)
  # make smaller margins
  par(mfrow=c(5,1), mar=c(3,2,2,1), oma=c(1,1,1,1))

hist(time2, breaks=1    , main="1 break"   , xlim=c(24.80,24.84), xlab=""); rug(time2)
hist(time2,               main="default"   , xlim=c(24.80,24.84), xlab=""); rug(time2)
hist(time2, breaks=10   , main="10 breaks" , xlim=c(24.80,24.84), xlab=""); rug(time2)
hist(time2, breaks=20   , main="20 breaks" , xlim=c(24.80,24.84), xlab=""); rug(time2)
hist(time2, breaks=100  , main="100 breaks", xlim=c(24.80,24.84), xlab=""); rug(time2)

  # restore par() settings
  par(old.par)
```

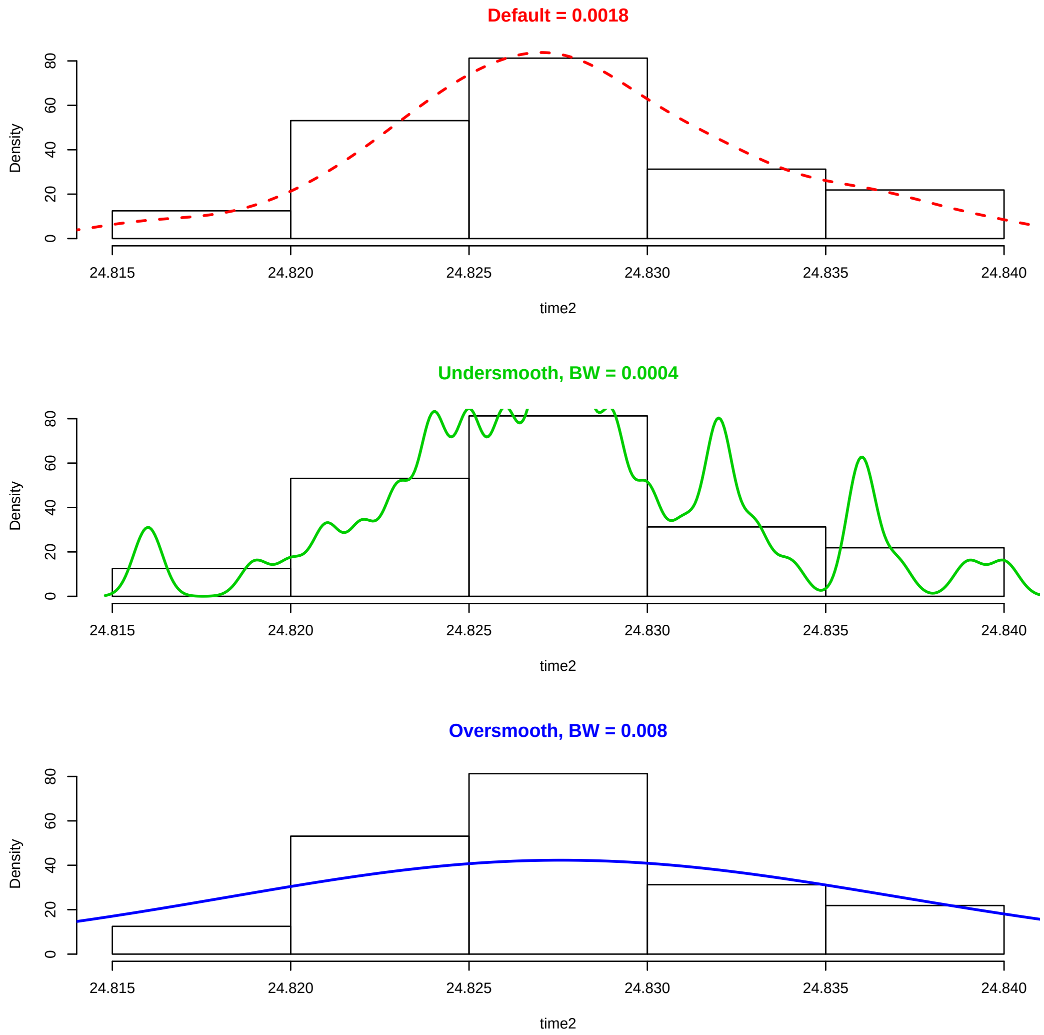


Notice that we are starting to see more and more bins that include only a single observation (or multiple observations at the precision of measurement). Taken to its extreme, this type of exercise gives in some sense a "perfect" fit to the data but is useless as an estimator of shape.

On the other hand, it is obvious that a single bin would also be completely useless. So we try in some sense to find a middle ground between these two extremes: "Oversmoothing" by using only one bin and "undersmooting" by using too many. This same paradigm occurs for density estimation, in which the amount of smoothing is determined by a quantity called the bandwidth. By default, R uses an optimal (in some sense) choice of bandwidth.

We've already used the `density()` function to provide a smooth curve to our histograms. So far, we've taken the default "bandwidth". Let's see what happens when we use different bandwidths.

```
par(mfrow=c(3,1))

# prob=TRUE scales the y-axis like a density function, total area = 1
hist(time2, prob=TRUE, main="")
# apply a density function, store the result
den = density(time2)
# plot density line over histogram
lines(den, col=2, lty=2, lwd=2)
# extract the bandwidth (bw) from the density line
b = round(den$bw, 4)
title(main=paste("Default =", b), col.main=2)

# undersmooth
hist(time2, prob=TRUE, main="")
lines(density(time2, bw=0.0004), col=3, lwd=2)
text(17.5, .35, "", col=3, cex=1.4)
title(main=paste("Undersmooth, BW = 0.0004"), col.main=3)

# oversmooth
hist(time2, prob=TRUE, main="")
lines(density(time2, bw=0.008), col=4, lwd=2)
title(main=paste("Oversmooth, BW = 0.008"), col.main=4)
```



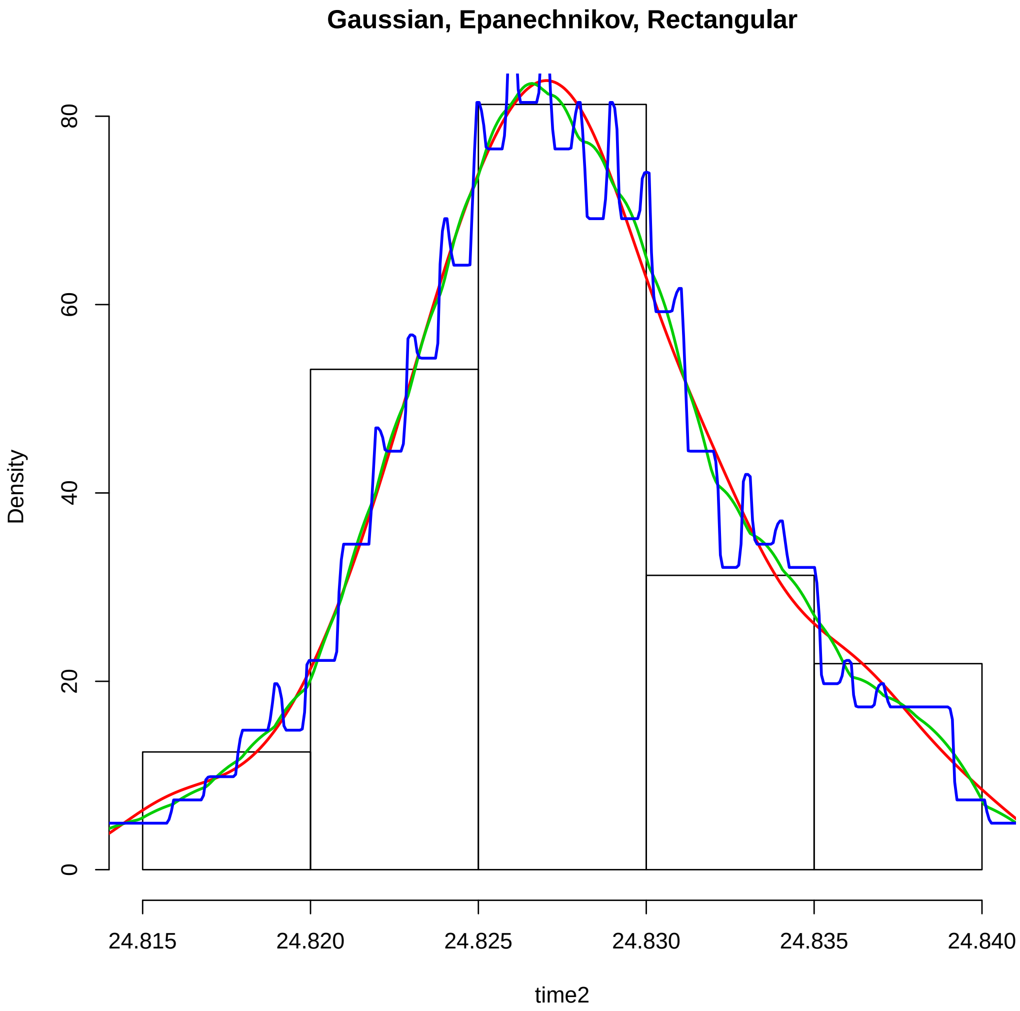
Default = 0.0018

Undersmooth, BW = 0.0004

Oversmooth, BW = 0.008

The other determining factor is the kernel, which is the shape each individual point takes before all the shapes are added up for a final density line. While the choice of bandwidth is very important, the choice of kernel is not. Choosing a kernel with hard edges (such as ”rect”) will result in jagged artifacts, so smoother kernels are often preferred.

```
par(mfrow=c(1,1))

hist(time2, prob=TRUE, main="")

# default kernel is Gaussian ("Normal")
lines(density(time2)             , col=2, lty=1, lwd=2)
lines(density(time2, ker="epan"), col=3, lty=1, lwd=2)
lines(density(time2, ker="rect"), col=4, lty=1, lwd=2)
title(main="Gaussian, Epanechnikov, Rectangular")

# other kernels include: "triangular", "biweight", "cosine", "optcosine"
```


Gaussian, Epanechnikov, Rectangular
Density
0
20
40
60
80
24.815
24.820
24.825
24.830
24.835
24.840
time2

Part IV
Additional topics