Displays for Statistics 5303

Lecture 13

October 2, 2002

Christopher Bingham, Instructor

612-625-7023 (St. Paul)
612-625-1024 (Minneapolis)

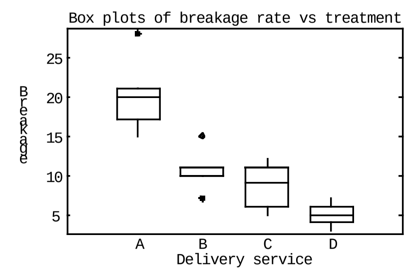
Class Web Page

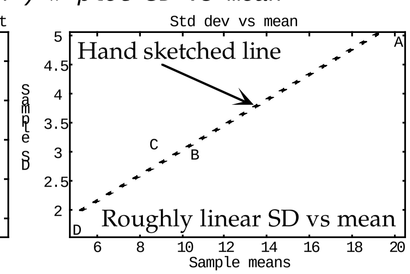
http://www.stat.umn.edu/~kb/classes/5303

Exercise 6.4

Treatment is choice of one of four over-night delivery services, A, B, C or D.
The response is breakage rate (percent).

```
Cmd> readdata("",treat,breakage)
Read from file "TP1:Stat5303:Data:Ch06:ex6-4.dat"
Column 1 saved as REAL vector treat
Column 2 saved as REAL vector y

Cmd> treat[run(5)] # check I got columns correctly
(1)           1           1           1           1           1

Cmd> treat <- factor(treat) # turn into factor

Cmd> list(treat,breakage) # see what we have
breakage        REAL   20
treat           REAL   20    FACTOR with 4 levels

Cmd> stats <- tabs(breakage,treat,mean:T,stddev:T)

Cmd> vboxplot(split(breakage,treat),ylab:"Breakage",\
 xlab:"Delivery service",xticklab:vector("A","B","C","D"),\
 title:"Box plots of breakage rate vs treatment")

Cmd> plot(stats$mean,stats$stddev,\
       symbols:vector("A","B","C","D"),\
       xlab:"Sample means",ylab:"Sample SD",\
       title:"Std dev vs mean") # plot SD vs mean
```



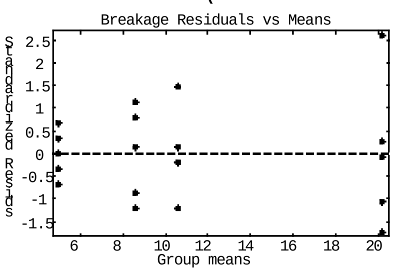


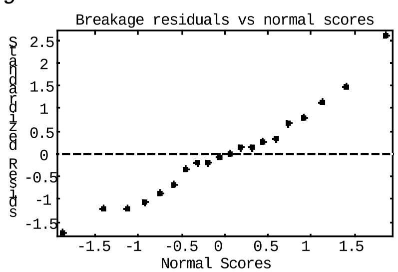
Clearly σ differs among groups, possibly related linearly to the mean.

The Linearity of the plot of SD vs mean suggests a log transform may be useful.

Make a couple of residual plots.

```
Cmd> anova("breakage=treat") # must precede resvsxxxx()
Model used is breakage=treat
                DF         SS         MS
CONSTANT         1     2464.2     2464.2
treat            3      632.6     210.87
ERROR1          16      179.2       11.2

Cmd> resvsyhat(title:"Breakage Residuals vs Means",\
    xlab:"Group means")

Cmd> resvsrankits(title:"Breakage residuals vs normal scores")
```





The left plot of residuals vs $\overline{y_i}$ shows the same pattern as the boxplots: When μ is high, σ is bigger than when μ is low. This is more evidence of heteroskedas-ticity.

The right normal scores plot is pretty straight, not putting normality in doubt.

There is an objective way to judge whether the plot is curved enough to be evidence against normality.

Compute the Pearson correlation r of the normal scores and the ordered values of residuals or standardized residuals.

For a perfect straight line r = 1, and the less straight the smaller r is.

An objective test is to reject

H_0: residuals are normal

if r is "too small", that is if $r \leq r_\alpha$, where r_α is a lower tail probability point of the distribution of r: $P(r \leq r_\alpha) = \alpha$.

There are few if any tables available, but you can find an approximate value by simulation. Still better, you can est-imate a p-value by simulation. You generate many sets of data with normal residuals so that H_0 is true. For each set you find residuals and computer r.

Here is how you might do it with these data. I work with the standardized residuals because that is what was plotted.

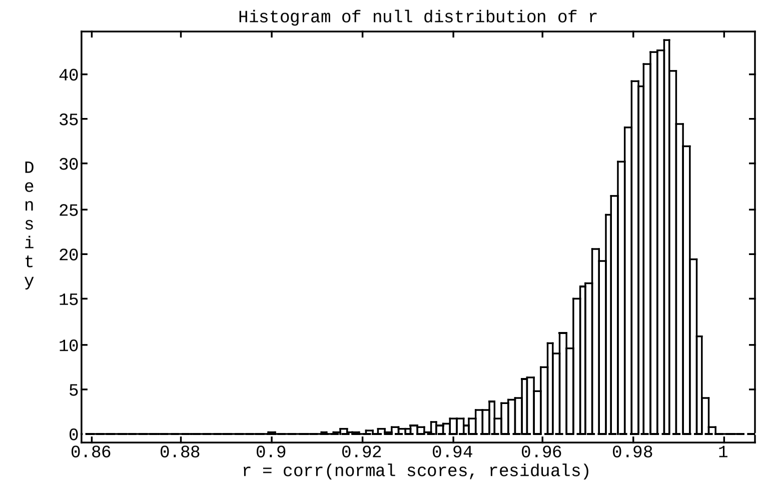
```
Cmd> M <- 5000;R <- rep(0,M) # simulate M samples

Cmd> normal_scrs <- rankits(n:20) # normal scores

Cmd> for(i,1,M){
     anova("{rnorm(20)} = treat",silent:T)
     mse <- SS[3]/DF[3]
     R[i] <- cor(normal_scrs,
        sort(RESIDUALS/(sqrt(1 - HII)*mse)))[1,2];;}
```

`RESIDUALS/(sqrt(1 - HII)*mse)` is the vector of internally standardized residuals. Now compute the observed r.

```
Cmd> anova("breakage=treat",silent:T)

Cmd> mse <- SS[3]/DF[3]

Cmd> r_observed <-\
     cor(normal_scrs,sort(RESIDUALS/(sqrt(1 - HII)*mse)))[1,2]

Cmd> r_observed # Could this be significantly low?
(1,1)     0.97966
```


Here is the simulated distribution of r for truly normal data.

```
Cmd> hist(R,100,title:"Histogram of null distribution of r",\
xlab:"r = corr(normal scores, residuals)")
```



The observed value r_{obs} = 0.97966 is clearly not unusual. Here is a estimated lower-tail P-value

```
Cmd> sum(R <= r_observed)/M # p-value
(1,1)      0.4496
```

and critical values

```
Cmd> J <- round(vector(.10,.05,.01,.001)*M); J
(1)         500         250          50           5

Cmd> sort(R)[J] # approx 10%, 5%, 1%, and 0.1% critical values
(1)      0.96147     0.95295     0.93035     0.89905
```

A little more on the Box-Cox transformation. On Monday I defined the Box-Cox transformation for power p to be

$y \to (y^p - 1)/p$ when $p \neq 0$
$y \to \log(y)$ when $p = 0$

The *geometric mean* GM of $y_1, \ldots, y_N$ is $GM \equiv e^{\overline{\log y}} = e^{(\sum \log y_i)/N}$.

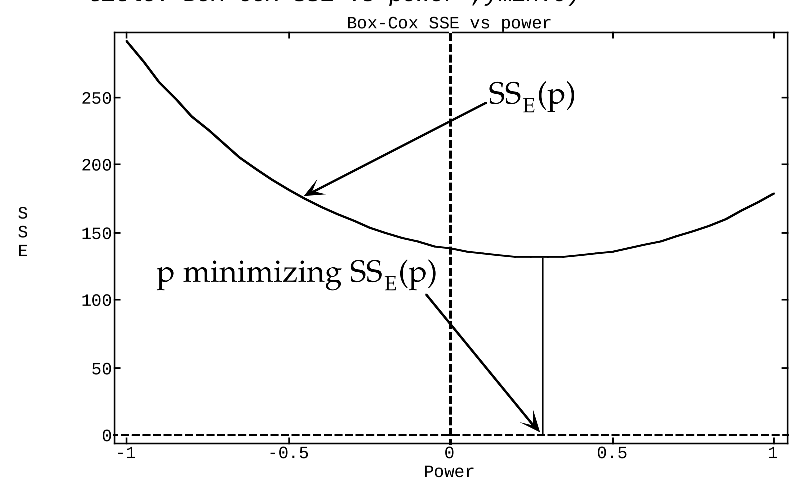
Oehlert defines the Box-Cox transformation similarly, except the transformed value is divided by GM^{p-1}:

$y \to y^{(p)} \equiv \{(y^p - 1)/p\}/GM^{p-1}$ $p \neq 0$
$y \to y^{(0)} \equiv GM \times \log(y)$ $p = 0$

This has the result that no matter what p is, the scale of the transformed values is comparable and indeed is in the same units as y. This means that all $SS_E(p)$ computed from $y^{(p)}$ are comparable. The value of p that minimizes $SS_E(p)$ is often a good transformation.

I'm going to use **boxcoxvec()** to try to select a transformation. This runs **anova()** using $y^{(p)}$ and returns a vector containing $SS_E(p)$ for several powers p.

```
Cmd> stuff <- boxcoxvec("treat",breakage,power:run(-1,1,.05))
  WARNING: searching for unrecognized macro boxcoxvec near
  stuff <- boxcoxvec(

  Cmd> compnames(stuff)
  (1) "power"
  (2) "SS"

  Cmd> lineplot(Power:stuff$power,SSE:stuff$SS,\
           title:"Box-Cox SSE vs power",ymin:0)
```



Where is the minimum?

```
Cmd> jmin <- grade(stuff$SS)[run(3)]; jmin
(1)          27          26          28
```

These are the indices of the three smallest values of `stuff$SS`.

```
Cmd> hconcat(stuff$power,stuff$SS)[vector(26,27,28),]
(1,1)         0.25      132.09
(2,1)          0.3      132.06
(3,1)         0.35      132.42
```

The minimum of the compute values of $SS_E(p)$ was for p = .3. This might suggest a cube root (p=1/3 = .3333). Or, since .3 is not very far from 0 or from .5, it might suggest a log or square root transformation.

What values of p are consistent with the data?

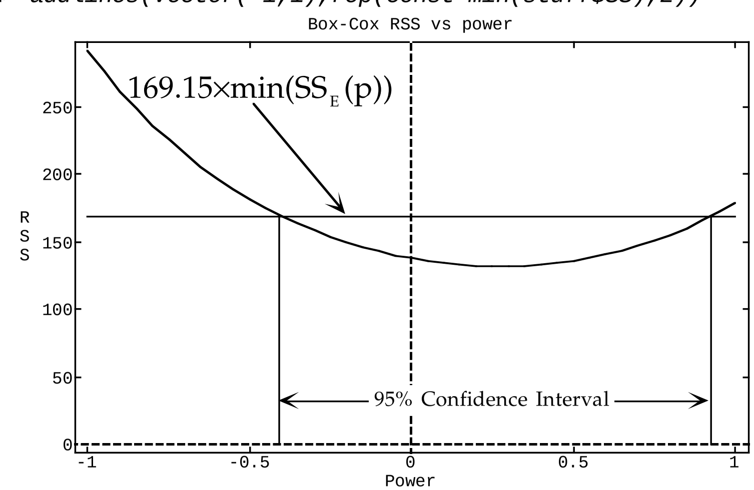
An approximate 1-α confidence interval for the "correct" p is the set of all powers p such that

$$SS_E(p) \le \min_p SS_E(p)\times(1 + F_{\alpha,1,df_{error}}/df_{error})$$

```
Cmd> const <- 1 + invF(1 - .05, 1, DF[3])/DF[3]; const
(1)      1.2809

Cmd> const*min(stuff$SS)
(1)      169.15
```

You can't exclude any p for which $SS_E(p) \le 1.2809\times132.06 = 169.15$.

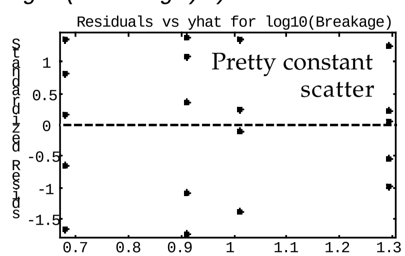
```
Cmd> addlines(vector(-1,1),rep(const*min(stuff$SS),2))
```

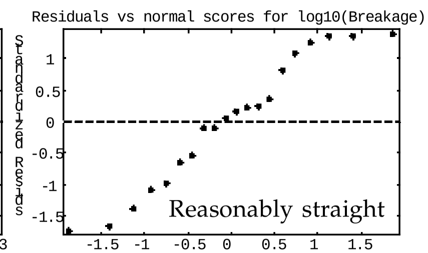


Arrows and annotations added by hand. p = 0, 1/3, 1/2 are in interval but not 1.

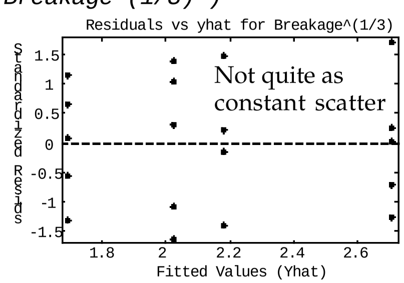
Look at residuals of transformed data:

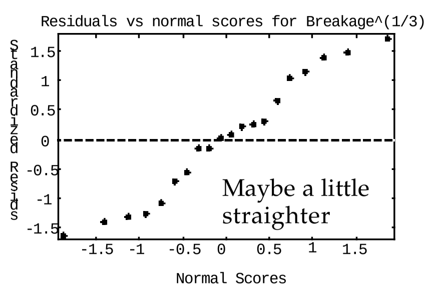
```
Cmd> anova("{log10(breakage)} = treat",fstat:T) # for log(y)
Model used is {log10(breakage)} = treat
              DF          SS          MS           F     P-value
CONSTANT       1      18.999      18.999  1039.54802           0
treat          3     0.97545     0.32515    17.79136  2.3827e-05
ERROR1        16     0.29241    0.018276

Cmd> resvsyhat(title:"Residuals vs yhat for log10(Breakage)")

Cmd> resvsrankits(title:"Residuals vs normal scores for
log10(Breakage)")
```






```
Cmd> anova("{breakage^(1/3)}=treat",fstat:T) # cube root
Model used is cuberoot=treat
              DF          SS          MS           F     P-value
CONSTANT       1      92.664      92.664  2010.90670           0
treat          3      2.6894     0.89648    19.45466  1.3787e-05
ERROR1        16     0.73729    0.046081

Cmd> resvsyhat(title:"Residuals vs yhat for Breakage^(1/3)")

Cmd> resvsrankits(title:"Residuals vs normal scores for
Breakage^(1/3)")
```





There are other ways to choose transformations:

Regression of log(SD) on log(mean): p = 1 - $\hat{\beta}$ is a guess at a good power to stabilize σ.

```
Cmd> regress("{log(stats$stddev)} = {log(stats$mean)}")
Model used is {log(stats$stddev)} = {log(stats$mean)}
                        Coef      StdErr           t
CONSTANT            -0.74709     0.30833      -2.423
{log(stats$mean)}    0.79133     0.13199      5.9954

N: 4, MSE: 0.017408, DF: 2, R^2: 0.94729
Regression F(1,2): 35.945, Durbin-Watson: 2.9443
To see the ANOVA table type 'anova()'

Cmd> COEF # automatically created by regress()
    CONSTANT {log(stats$mean)}
    -0.74709     0.79133    Intercept and slope

Cmd> slope <- COEF[2] # slope is second coefficient

Cmd> 1 - slope # guess of power
(1)     0.20867
```

p = 0.209 is in the same ballpark as was found using **boxcoxvec()**.

Note: You seldom, if ever, use the exact value found by **boxcoxvec()** or this regression method. You usually pick a "neat" value such as p = -1, 0, 1/3 or 1/2.

In some cases, some math can suggest a transformation which will stabilize σ.

Suppose you are trying to find a transformation $y \rightarrow \tilde{y} \equiv f(y)$.

If $f(y)$ is a smooth monotonic (always increasing or decreasing) function, it is not hard to show using the δ-method that

$$\sigma_{\tilde{y}}^2 \cong (f'(\mu))^2 \sigma_y^2$$

where $f'(\mu)$ is the derivative of $f(\mu)$.

Now suppose σ_y^2 depends on $\mu = \mu_y$, say

$$\sigma_y^2 = \sigma(\mu)^2 = g(\mu)$$

Then

$$\sigma_{\tilde{y}}^2 \cong (f'(\mu))^2 g(\mu)$$

If you want this to be constant, K^2, then use $f(y)$ such that

$$f'(y) = K/\sqrt{g(y)}$$

This is a differential equation that can be solved for $f(y)$ in some cases.

Examples

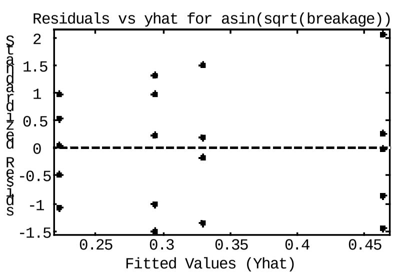
- $\sigma(\mu)^2 = g(\mu) = C\mu$
  $f(\mu) = k\sqrt{\mu}$, i.e., square root
  When y is Poisson, $\sigma(\mu)^2 = \mu$
  The Poisson distribution is a distribution for counts with $P(y=k) = e^{-\mu}\mu^k/k!$
- $\sigma(\mu)^2 = g(\mu) = C\mu^2$
  $f(\mu) = k \times \log \mu$,
  This applies when y is Gamma or χ^2
- $\sigma(\mu)^2 = g(\mu) = C\mu(1 - \mu)$
  $f(\mu) = \sin^{-1}(\sqrt{\mu})$
  This applies when $y = \hat{p} = X/n$, where X is binomial.

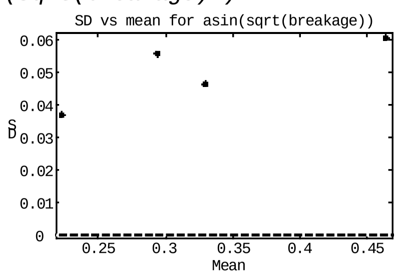
  Even with non-binomial data, $\sin^{-1}(\sqrt{y})$ is often tried when y is a proportion or percent ($\sin^{-1}(\sqrt{\{percent/100\}})$)

Note: $\sin^{-1}(x)$ satisfies $\sin(\sin^{-1}(x)) = x$.

The MacAnova function **asin(x)** computes $\sin^{-1}(x)$.

```
Cmd> sin(asin(.123)) # sin(asin(x)) is x for -1 ≤ x ≤ 1
(1)       0.123

Cmd> y1 <- asin(sqrt(breakage/100))

Cmd> anova("y1=treat",fstat:T)
Model used is y1=treat
               DF         SS           MS           F      P-value
CONSTANT        1     2.1469       2.1469   829.11074            0
treat           3    0.15317     0.051057    19.71791   1.2686e-05
ERROR1         16    0.04143    0.0025894

Cmd> stats <- tabs(y1,treat,mean:T,stddev:T)

Cmd> resvsyhat(title:"Residuals vs yhat for
asin(sqrt(breakage))")

Cmd> plot(Mean:stats$mean,SD:stats$stddev,ymin:0,\
    title:"SD vs mean for asin(sqrt(breakage)")
```





The plots show some remaining hetero-skedasticity.

Comment. For small p, $\sin^{-1}\sqrt{p} \cong \sqrt{p}$, so the transformation is like a square root.

Sometimes when the variances differ between groups you don't want to work with a transformation because

- the original scale has some special importance

or

- you can't find a good transformation

There are approximate ANOVA or t-test methods available, which don't work with single pooled estimate of variance

The variance of a contrast $\sum_i w_i \overline{y_{i\bullet}}$ is

$$V[\sum_i w_i \overline{y_{i\bullet}}] = \sum_i w_i^2 \sigma_i^2/n_i$$

So an estimate of the standard error is

$$\hat{SE}[\sum_i w_i \overline{y_{i\bullet}}] = \sqrt{\{\sum_i w_i^2 s_i^2/n_i\}}$$

The "t-statistic" to test $H_0: \sum_i w_i \alpha_i = 0$

$$t_w = \sum_i w_i \overline{y_{i\bullet}} / \sqrt{\{\sum_i w_i^2 s_i^2/n_i\}}$$

does not have Student's t-distribution, but t_ν is a good approximation when

$$\nu = \{\sum_i w_i^2 s_i^2/n_i\}^2/\{\sum_i w_i^4 s_i^4/((n_i-1)n_i^2)\}$$

Here I illustrate it comparing the first delivery services A and B with C and D:

```
Cmd> stats <- tabs(breakage,treat)

Cmd> vars <- stats$var; vars # sample variances s_i^2
(1)       24.7         8.3         9.3         2.5

Cmd> n <- stats$count; n # sample sizes
(1)          5           5           5           5

Cmd> ybars <- stats$mean; ybars # sample means
(1)       20.2        10.6         8.6           5

Cmd> w <- vector(1,1,-1,-1) # contrast weights

Cmd> estimate <- sum(w*ybars); estimate # of contrast
(1)       17.2

Cmd> se <- sqrt(sum(w^2*vars/n)); se # std error of contrast
(1)     2.9933

Cmd> tstat <- estimate/se; tstat # test statistic
(1)     5.7461

Cmd> df <- sum(w^2*vars/n)^2/sum(w^4*vars^2/((n-1)*n^2)); df
(1)     10.403      Approximate d.f.

Cmd> twotailt(tstat,df) # Approximate P-value
(1)  0.00016003      Reject H_0.
```


The Brown-Forsythe test is a modification of the ANOVA F-test.

Define

$$d_i = s_i^2(1 - n_i/N) = s_i^2((N - n_i)/N)$$

Then the statistic is

$$BF \equiv SS_{trt}/\sum_i d_i$$

$SS_{trt} = \sum_i n_i(\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2$ is the usual ANOVA treatment SS.

When H_0: $\alpha_1 = \ldots = \alpha_g$ is true, BF is is distributed approximately as F on g-1 and ν d.f., where

$$\nu = (\sum d_i)^2/\sum(d_i^2/(n_i-1))$$

When $n_1 = n_2 = \ldots = n_g = n$,

$\sum_i d_i = ((g-1)/g)\sum s_i^2 = (g-1)MS_E$ so BF = F.

This puts a premium on having equal sample sizes, since F is also BF, but with smaller denominator degrees of freedom.

Here I found SS_{trt}, the numerator of BF, by a "white box" method:

```
Cmd> grandmean <- describe(breakage,mean:T)

Cmd> top <- sum(n*(ybars-grandmean)^2); top
(1)       632.6

Cmd> bottom <- sum(vars*(1-n/sum(n))); bottom

Cmd> bf <- top/bottom; bf
(1)      18.827

Cmd> d <- vars*(1 - n/sum(n))

Cmd> df <- sum(d)^2/sum(d^2/(n-1)); df
(1)      10.403

Cmd> 1 - cumF(bf,3,df)
(1)  0.00016065
```

Here is the ordinary ANOVA.

```
Cmd> anova("breakage=treat",fstat:T)
Model used is breakage=treat
              DF          SS          MS           F      P-value
CONSTANT       1      2464.2      2464.2   220.01786      < 1e-08
treat          3       632.6      210.87    18.82738  1.6873e-05
ERROR1        16       179.2        11.2
```

F is the same as BF, but has 16 denominator d.f. instead of 10.4. The P-value is smaller but the conclusion is the same.