Kruskal Wallis Test for Differences in Distribution

Note on required packages: The following code required the package `psych` to perform statistics related to the median. If you have not already done so, download, install, and load the library with the following code:

```
install.packages("psych")
```

```
library("psych")
```

---

1. Introduction

Kruskal Wallis Test is a statistical method to determine if there is a difference in distributions between two or more *independent* groups, where the groups are defined by the outcomes for a single categorical variable.

Kruskal Wallis Test can be assumed to test the hypothesis that the *center* (as measured by the median or interpolated median) of the distribution for the outcome variable is different for different categories of the explanatory variable, if it can be assumed that the *shape of the distribution* is the same across categories.

In its purpose, it is essentially an non-parametric alternative of the one-way ANOVA for a difference in means for two or more groups.

Because the Kruskal Wallis test is a non-parametric test using ranks of the observations in its calculations, it can be performed on *ordinal* data or interval/ratio data.

2. Hypotheses

The null hypothesis says that the center of all the groups are equal to each other. The alternative hypothesis says that *at least one* of the categories has a different measure of center than the others.

Let $K \geq 2$ denote the number of groups for the explanatory variable. Let M_i denote the population interpolated median for group i. The null and alternative hypotheses for the Kruskal Wallis Test are given by,

Null: $M_1 = M_2 = ... = M_K$

Alternative: $M_i \neq M_j$ for some $i \neq j$

3. Assumptions

The following assumptions are necessary for the Kruskal Wallis test:

- **Random assignment:** Observations are randomly assigned to the K groups independently of the outcome variable.
- **Independent groups:** Observations in one group are independent of observations in all other groups. There must therefore be different sampling units in each group.
- **Homogeneity in distribution shape:** The shape of the distribution of the outcome variable for the different groups are the same, even though the population medians or interpolated medians may be different.

4. Example: Current Population Survey

The data set below includes data from more than 52,000 individuals over the age of 25 years that participated in the 2016 Current Population Survey. The variables include the following:

- `HOURS`: Usual weekly hours worked over all jobs (ratio)
- `SEX`: Male or female (categorical)
- `RACE`: Racial identity (categorical)
- `EDU`: Education (categorical)
- `EDUNUM`: A numerical code for the education categories in `EDU` where larger numbers indicate a higher level of educational attainment
- `MARRIED`: Marital status (categorical)

The following code downloads the data, opens it, and saves it in a dataframe that we call `wdat`.

```
wdat <- read.csv("http://murraylax.org/datasets/cpshours.csv")
```

We can use this data to determine if educational attainment, an ordinal variable, is different for people in different categories for one or more of other the categorical variables.

5. Computing Medians / Interpolated Medians

Let us calculate the median and interpolated median level of education for one of the racial groups, say those who identified themselves as "White". The following estimates the sample median and interpolated median, respectively.

```
median( wdat$EDUCNUM[ wdat$RACE=="White" ] )
```

```
## [1] 3
```

```
interp.median( wdat$EDUCNUM[ wdat$RACE=="White" ] )
```

```
## [1] 2.570952
```

The parameter to each of these functions is the variable `wdat$EDUCNUM`, but only the subset where `wdat$RACE` is equal to "White." The square brackets, [], are used to specify which rows of `wdat$RACE` to select. Specifically, we specify only the observations where the expression `wdat$RACE=="White"` is equal to TRUE.

We can make general functions that allow us to call these procedures more easily. The following code creates a function called `median.race` and sets it equal to the function above, where `x` can equal the text describing any race the user wishes to supply.

```
median.race <- function(x) median(wdat$EDUCNUM[wdat$RACE==x])
```

Similarly, we can make an interpolated median function, and call it `interp.median.race`:

```
interp.median.race <- function(x) interp.median(wdat$EDUCNUM[wdat$RACE==x])
```

Now we can estimate the median and interpolated median for white people with the following calls:

```
median.race("White")
```

```
## [1] 3
```

```
interp.median.race("Black")
```

```
## [1] 2.353535
```

We can view all the possible racial categories with the following call to `unique()`:

```
races<-unique(wdat$RACE)
races
```

```
## [1] White                          Asian/Pacific Islander
## [3] Other                          Black
## [5] American Indian/Aleut/Eskimo
## 5 Levels: American Indian/Aleut/Eskimo Asian/Pacific Islander ... White
```

Finally, we can repeatedly call our functions `median.race()` and `interp.median.race()` over all the possible outcomes for race with the following call to `sapply()`:

```
meds <- sapply(X=races, FUN=median.race)
imeds <- sapply(X=races, FUN=interp.median.race)
```

The calls above saved the medians and interpolated medians in the objects `meds` and `imeds`. We can see the values of the medians and interpolated medians by typing in the names of these objects at the command prompt:

```
meds
```

```
## [1] 3 3 2 2 2
```

```
imeds
```

```
## [1] 2.570952 3.021533 2.392857 2.353535 2.140299
```

The output is not too useful as it requires you to remember the racial categories and how they were ordered. We can *name* the elements of these lists by their racial category with a call to `names()`.

```
names(meds) <- races
names(imeds) <- races
```

The code above assigns the names of `meds` and `imeds` to the strings in the list, `races`.

And now again we can view the contents of these objects in two nicely labeled columns.

```
cbind(meds,imeds)
```

```
##                               meds    imeds
## White                            3 2.570952
## Asian/Pacific Islander           3 3.021533
## Other                            2 2.392857
## Black                            2 2.353535
## American Indian/Aleut/Eskimo     2 2.140299
```

The `cbind()` function stands for "column bind" and simply binds the columns together.

We can see there are differences in the sample medians and interpolated medians for level of education between different races. In the next section, we formally test for a difference in medians.

6. Conducting the Kruskal Wallis Hypothesis Test

The code below calls the `kruskal.test()` function to test the hypothesis that average educational level is different for people in different races.

```
kt <- kruskal.test(EDUCNUM ~ factor(RACE), data=wdat)
```

The *formula*, `EDUCNUM ~ factor(RACE)` tells the function to examine the outcome variable `EDUCNUM` as it depends on `RACE`.

Inside the formula is a call to the function, `factor()`. The `kruskal.test()` function requires the explanatory variable to have a limited number of categories and that these categories be numerical. The categorical variable `RACE` is not numerical, but rather each variable takes on a string describing the racial identification. The `factor()` function converts the the string factors to numerical factors.

The second parameter, `data=wdat`, specifies where the `kruskal.test()` function can to find the variables used in the preceding formula.

We can review the outcome of the statistical test by viewing the return object `kt`.

```
kt
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  EDUCNUM by factor(RACE)
## Kruskal-Wallis chi-squared = 783.79, df = 4, p-value < 2.2e-16
```

With a p-value equal to 2.2×10^{-16}, we have sufficient statistical evidence that the average level of educational attainment is different for different racial groups.