Heatmaps for Economic Analysis

Tom Cui, Eric Zwick
(DRAFT)

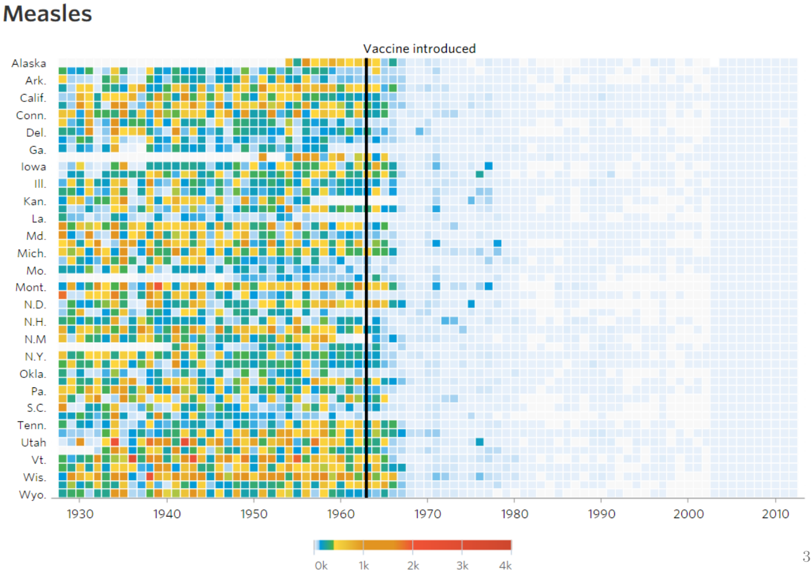
September 6, 2016

What is a heatmap?

- **A two-dimensional visualization of data using colour to represent magnitude**
- Broad definition, which could be divided into
- **Embedded** heatmaps that overlay colour on an actual map or image (not covered here)
- **Matrix** heatmaps that presents a grid of values where colours differ by cell

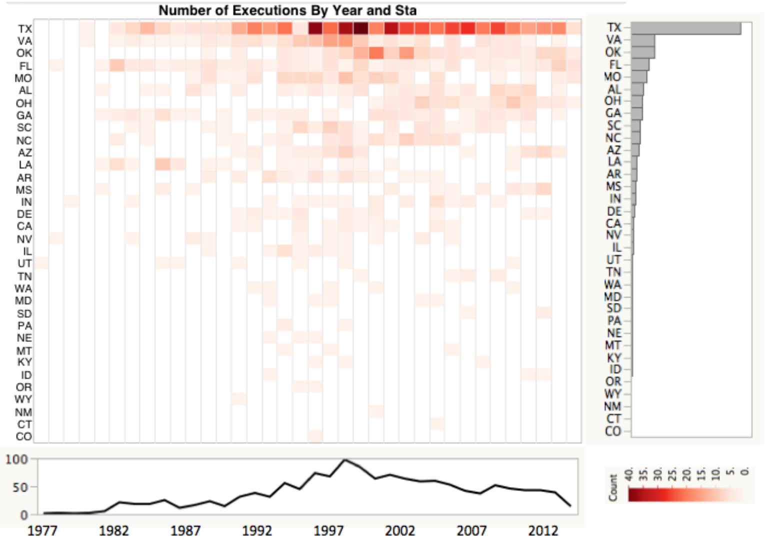
What is a heatmap?

Example: The WSJ vaccine visualization (DeBold, Friedman 2015)



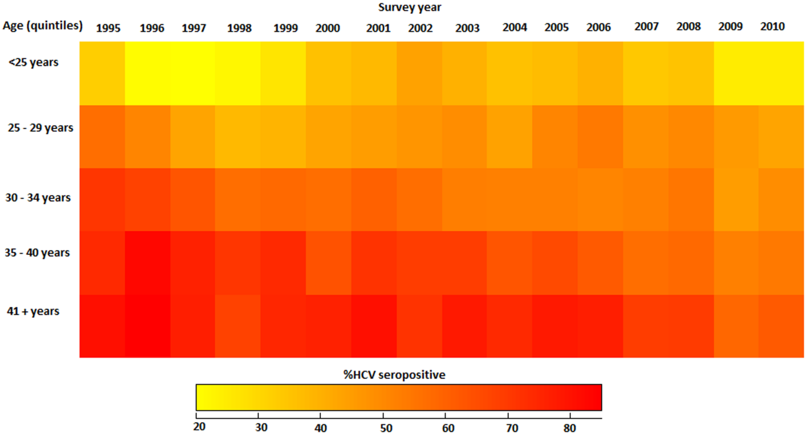
What is a heatmap?

Example: Kaiser Fung's executions data



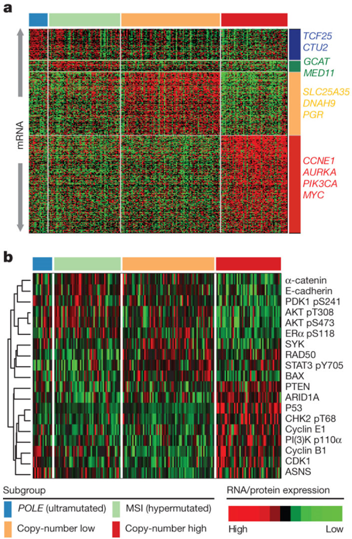
What is a heatmap?

Example (Bad): A "quilt plot" of Hep C prevalence (Wand et al)



What is a heatmap?

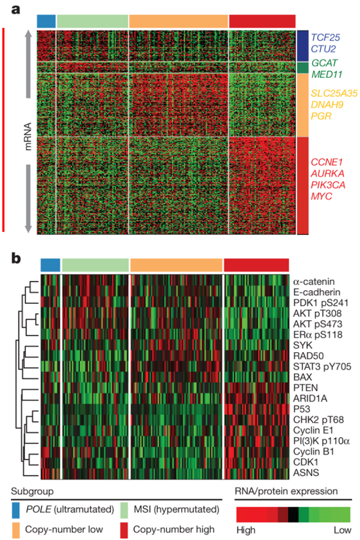
Example: Plotting gene expression data over samples (TCGN 2013)



What is a heatmap?

Example: Plotting gene expression data over samples (TCGN 2013)

Each row ($\sim$ 1500) is one gene

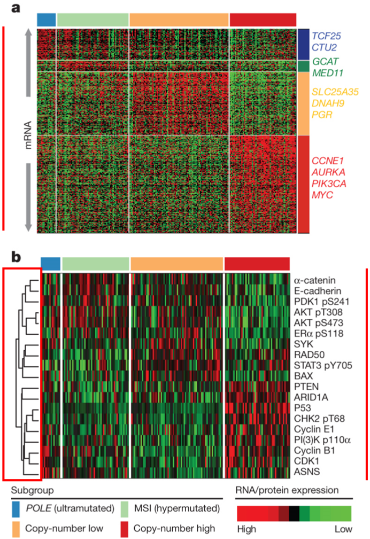


What is a heatmap?

Example: Plotting gene expression data over samples (TCGN 2013)

Each row ($\sim$ 1500) is one gene

Dendrogram



Each row is a protein

What is a heatmap?

Some takeaways from these examples:

- The axes change the interpretation
  (1) - (3) use time as the X and factors as the Y, (4) uses factors for both

What is a heatmap?

Some takeaways from these examples:

- The axes change the interpretation
  (1) - (3) use time as the X and factors as the Y, (4) uses factors for both
- Good representation of high-dimensional data
  (4) is an extreme example of this, but common in bioinformatics

What is a heatmap?

Some takeaways from these examples:

- The axes change the interpretation
  (1) - (3) use time as the X and factors as the Y, (4) uses factors for both
- Good representation of high-dimensional data
  (4) is an extreme example of this, but common in bioinformatics
- Permuting axis order improves interpretation
  (2) sorts Y by total count over the sampling period, (4) uses cluster analysis (recall dendrogram)

Setting up a heatmap for economics

- In an ideal world, we could derive causal effects in a model $Y = g(W)$ using exogeneous assignment of W and observing the entire support of W

Setting up a heatmap for economics

- In an ideal world, we could derive causal effects in a model $Y = g(W)$ using exogeneous assignment of W and observing the entire support of W
- Big data makes the latter easier. Former still hard!
- Hence research designs that exploit a policy introduction or kink are popular

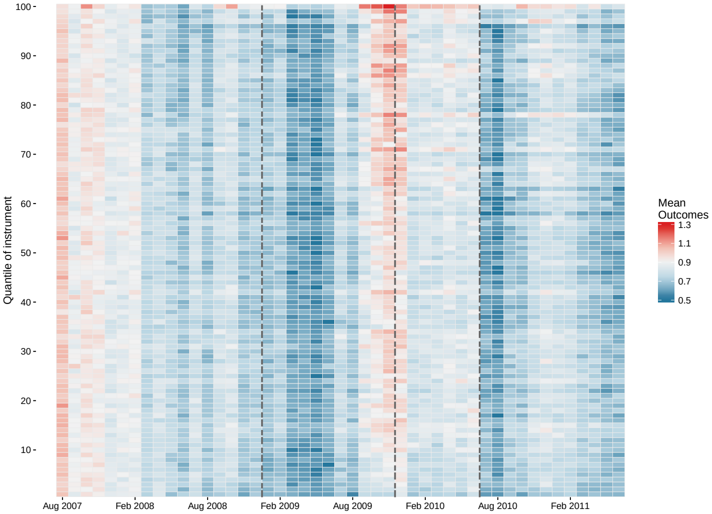
Setting up a heatmap for economics

- In an ideal world, we could derive causal effects in a model $Y = g(W)$ using exogeneous assignment of W and observing the entire support of W
- Big data makes the latter easier. Former still hard!
- Hence research designs that exploit a policy introduction or kink are popular

Now consider a heatmap where time is on the X axis (**showing the policy introduction**) and where W, or a variable related to a latent W, (**showing the support of W**) is binned on the Y axis

Setting up a heatmap for economics

Example: Scaled house sales in a heatmap sorted by FTHB exposure, from Berger, Turner, Zwick ()



Setting up a heatmap for economics

Using earlier takeaways:

- The axes change the interpretation
  Placing time on X and an instrument of W on Y implies this heatmap is a visualization of nonparametric regression

Setting up a heatmap for economics

Using earlier takeaways:

- The axes change the interpretation
  Placing time on X and an instrument of W on Y implies this heatmap is a visualization of nonparametric regression
- Good representation of high-dimensional data
  Around 8600 ZIPs binned into 100 percentiles

Setting up a heatmap for economics

Using earlier takeaways:

- The axes change the interpretation
  Placing time on X and an instrument of W on Y implies this heatmap is a visualization of nonparametric regression
- Good representation of high-dimensional data
  Around 8600 ZIPs binned into 100 percentiles
- Permuting axis order improves interpretation
  Y axis sorted to be increasing in the instrument of W, and figure tells us the effect of W on Y is positive in a linear model

Setting up a heatmap for economics

Extensions:

- Time on X, other variables on Y, plotting means
  $=$ **Covariate balance check**

Setting up a heatmap for economics

Extensions:

- Time on X, other variables on Y, plotting means
  = **Covariate balance check**
- Time on X, individual stocks on Y, plotting market-adjusted returns
  = **Financial event study**

Setting up a heatmap for economics

Extensions:

- Time on X, other variables on Y, plotting means
  = **Covariate balance check**
- Time on X, individual stocks on Y, plotting market-adjusted returns
  = **Financial event study**
- Time on X, generation on Y, plotting average of a simulated policy function
  = **OLG model dynamics**

Setting up a heatmap for economics

Extensions:

- Time on X, other variables on Y, plotting means
  = **Covariate balance check**
- Time on X, individual stocks on Y, plotting market-adjusted returns
  = **Financial event study**
- Time on X, generation on Y, plotting average of a simulated policy function
  = **OLG model dynamics**
- Policy-relevant index on X, quantiles of Y on Y, plotting obs. counts in bin
  = **Fuzzy RDD**

Setting up a heatmap for economics

Extensions:

- Time on X, other variables on Y, plotting means
  = **Covariate balance check**
- Time on X, individual stocks on Y, plotting market-adjusted returns
  = **Financial event study**
- Time on X, generation on Y, plotting average of a simulated policy function
  = **OLG model dynamics**
- Policy-relevant index on X, quantiles of Y on Y, plotting obs. counts in bin
  = **Fuzzy RDD**

and so on.

The heatmapEco package

The heatmapEco package

- **Many** programs for creating heatmaps exist

So why another package?

The heatmapEco package

- **Many** programs for creating heatmaps exist
  - Stata `twoway contour`, `hmap`
  - R base, `gplots`, `ggplot2`, `d3heatmap` ...
  - Matlab and Python `matplotlib`

So why another package?

The heatmapEco package

- **Many** programs for creating heatmaps exist
  - Stata `twoway contour`, `hmap`
  - R base, `gplots`, `ggplot2`, `d3heatmap` ...
  - Matlab and Python `matplotlib`

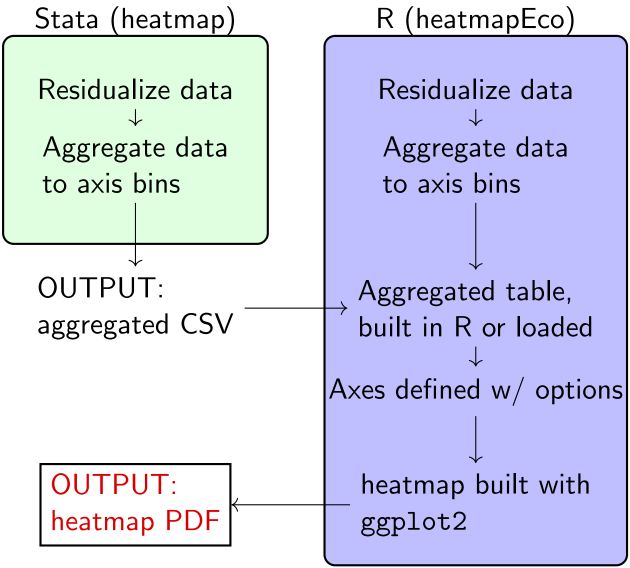
  So why another package?

- `heatmapEco` makes informative heatmaps easy by
  - **Focusing on proper design of axes;**
  - **Setting relevant axis permutations;**
  - **Completing prerequisite data cleaning.**

The heatmapEco package

- Complicated heatmaps like TCGN's are also quite uncomplicated; they are literally a projection of some tabular data
- In other words, the data loaded in is a 373x1500 matrix. The values are then standardized, variables are clustered and given a colour
- But instead data may need to be aggregated, reshaped; axes relabelled; colour palettes adjusted to show significant results
- `heatmapEco` combines R packages to simplify these changes and adds design features of its own

The heatmapEco package



heatmapEco axes

- Current support for X axis:
  - **Index axis** over numeric values (income, policy thresholds)
  - **Time axis** where time strings are converted into valid axis values by the package

heatmapEco axes

- Current support for X axis:
  - **Index axis** over numeric values (income, policy thresholds)
  - **Time axis** where time strings are converted into valid axis values by the package
- Current support for Y axis:
  - **Factor axis** where each entry is some (aggregated) grouping
  - **Quantile axis** where a continuous instrument is split by N quantiles

heatmapEco axes

- Current support for X axis:
  - **Index axis** over numeric values (income, policy thresholds)
  - **Time axis** where time strings are converted into valid axis values by the package
- Current support for Y axis:
  - **Factor axis** where each entry is some (aggregated) grouping
  - **Quantile axis** where a continuous instrument is split by N quantiles

Currently output is in landscape letter format, but ultimately axis placement should be arbitrary and portrait format heatmaps possible

heatmapEco aggregation

In R the aggregation process is inputted using a pseudo-formula

```
Y ~ CrS(X,ID,w):i(t)
```

where

- `Y` is the dependent variable, or the fill variable
- `X` is the factor independent variable or a continuous instrument to be binned
- `i` is the index or time axis
- `t` allows `X` to be sorted on its values at some time `t`, if `X` is time varying (**use caution**)
- `ID` is the individual identifier, either unique or unique with `t`
- `w` are quantile weights

heatmapEco aggregation

In R the aggregation process is inputted using a pseudo-formula

$$\texttt{Y} \sim \texttt{CrS(X,ID,w):i(t)}$$

where

- `Y` is the dependent variable, or the fill variable
- `X` is the factor independent variable or a continuous instrument to be binned
- `i` is the index or time axis
- `t` allows `X` to be sorted on its values at some time `t`, if `X` is time varying (**use caution**)
- `ID` is the individual identifier, either unique or unique with `t`
- `w` are quantile weights

In Stata the syntax is

```
heatmap Y X i [weights], id(varname) [t_sort(string)]
```

heatmapEco aggregation

- Note that, in R, an anonymous function could be passed as an argument. This means the aggregation function argument `grp.func` can take many forms, so long as a summary function is involved
- E.g. take the median of a quantile-month bin. Or take the log transform of that median. Or add control flow; if data censored, first remove censored data and output log median of what remains

heatmapEco aggregation

- Note that, in R, an anonymous function could be passed as an argument. This means the aggregation function argument `grp.func` can take many forms, so long as a summary function is involved
- E.g. take the median of a quantile-month bin. Or take the log transform of that median. Or add control flow; if data censored, first remove censored data and output log median of what remains
- Stata's aggregation features are much less rich: every collapse function could be inputted into `grpfunc`

heatmapEco residualization

Both dependent and independent variables can be first residualized according to a model

$$Y = \beta W + D\theta + F\psi + X\gamma + \varepsilon$$

Where D, F are fixed effects and X are controls.
Stata implementation uses base `areg`. R implementation uses `plm` or `lfe` (TODO)

Colour palettes

Standard divergent color palette

Semi-sequential palette for count data

- On standard palette, far two shades reserved for outlier detection: binned values above the 1.5 + IQR range are considerably darker
- Standard colors are not equally spaced: distribution below median take longer to get to dark blue hues. This is to emphasize “Ashenfelter dips”
- Count data palette is ColorBrewer YlOrBr, with high outliers and a muted hue to deemphasize data censored by 0 (by default)

heatmapEco Examples

WSJ replication

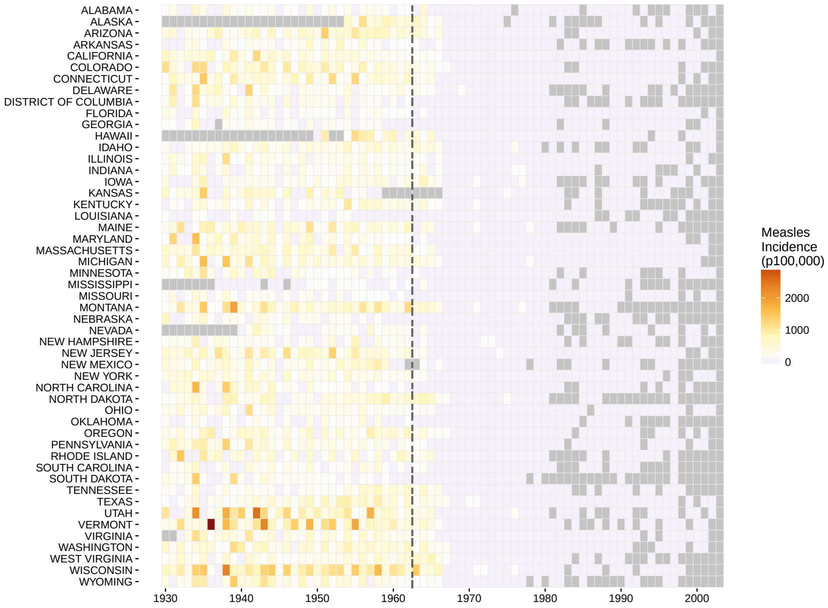
Download data from Project Tycho. The cleaning in R:

```
library(data.table)
obj <- melt(fread("MEASLES_Incidence_1930-2003.csv"),
                    c("YEAR", "WEEK"))
obj[, value := as.numeric(value)]
```

Calling heatmapEco:

```
nasum <- function(...)
         if (all(is.na(...))) NA else sum(..., na.rm=TRUE)
heatmapEco(value ~ CrS(variable,variable):YEAR, obj,
t.fmt="\%Y", t.per="year", pol.break=c("Jan 1963"),
grp.func=nasum, count=T, factor.ax=T, outliers=T, split.x=10,
zlab="Measles Incidence (p100,000)", save="measlesRep.pdf")
```

WSJ REPLICATION



WSJ replication

Line by line:

WSJ replication

Line by line:

- `heatmapEco(value ~ CrS(variable,variable):YEAR,obj,`
  Inputs formula for aggregation and dataset
- `t.fmt="%Y", t.per="year", pol.break=c("Jan 1963"),`
  Data object, time is in pure “year” format, policy line date

WSJ REPLICATION

Line by line:

- `heatmapEco(value ~ CrS(variable,variable):YEAR,obj,`
  Inputs formula for aggregation and dataset
- `t.fmt="%Y", t.per="year", pol.break=c("Jan 1963"),`
  Data object, time is in pure "year" format, policy line date
- `grp.func=nasum [nasum <- function(...)`
  `if (all(is.na(...)))  NA else sum(..., na.rm=TRUE)]`
  Grouping function is summation, excluding NAs (a year with NAs is inputted as NA, grayed out)

WSJ REPLICATION

Line by line:

- `heatmapEco(value ~ CrS(variable,variable):YEAR,obj,`
  Inputs formula for aggregation and dataset
- `t.fmt="%Y", t.per="year", pol.break=c("Jan 1963"),`
  Data object, time is in pure "year" format, policy line date
- `grp.func=nasum [nasum <- function(...)`
  `if (all(is.na(...)))  NA else sum(..., na.rm=TRUE)]`
  Grouping function is summation, excluding NAs (a year with NAs is inputted as NA, grayed out)
- `count=T, factor.ax=T, outliers=T, split.x=10,`
  Use the count colour palette; the Y-axis are state factors; turn on outlier perception; X tick every ten units

WSJ REPLICATION

Line by line:

- `heatmapEco(value ~ CrS(variable,variable):YEAR,obj,`
  Inputs formula for aggregation and dataset
- `t.fmt="%Y", t.per="year", pol.break=c("Jan 1963"),`
  Data object, time is in pure "year" format, policy line date
- `grp.func=nasum [nasum <- function(...)`
  `if (all(is.na(...)))  NA else sum(..., na.rm=TRUE)]`
  Grouping function is summation, excluding NAs (a year with NAs is inputted as NA, grayed out)
- `count=T, factor.ax=T, outliers=T, split.x=10,`
  Use the count colour palette; the Y-axis are state factors; turn on outlier perception; X tick every ten units
- `zlab="Measles Incidence (p100,000)",save="measlesRep.pdf")`
  Policy line, labels, output location.

WSJ REPLICATION

Line by line:

- `heatmapEco(value ~ CrS(variable,variable):YEAR,obj,`
  Inputs formula for aggregation and dataset
- `t.fmt="%Y", t.per="year", pol.break=c("Jan 1963"),`
  Data object, time is in pure "year" format, policy line date
- `grp.func=nasum [nasum <- function(...)`
  `if (all(is.na(...)))  NA else sum(..., na.rm=TRUE)]`
  Grouping function is summation, excluding NAs (a year with NAs is inputted as NA, grayed out)
- `count=T, factor.ax=T, outliers=T, split.x=10,`
  Use the count colour palette; the Y-axis are state factors; turn on outlier perception; X tick every ten units
- `zlab="Measles Incidence (p100,000)",save="measlesRep.pdf")`
  Policy line, labels, output location.

WSJ replication

Line by line:

- `heatmapEco(value ~ CrS(variable,variable):YEAR,obj,`
  Inputs formula for aggregation and dataset
- `t.fmt="%Y", t.per="year", pol.break=c("Jan 1963"),`
  Data object, time is in pure "year" format, policy line date
- `grp.func=nasum [nasum <- function(...)`
  `if (all(is.na(...)))  NA else sum(..., na.rm=TRUE)]`
  Grouping function is summation, excluding NAs (a year with NAs is inputted as NA, grayed out)
- `count=T, factor.ax=T, outliers=T, split.x=10,`
  Use the count colour palette; the Y-axis are state factors; turn on outlier perception; X tick every ten units
- `zlab="Measles Incidence (p100,000)",save="measlesRep.pdf")`
  Policy line, labels, output location.

Overall: **9 lines of code w/ data.table**

- **9 lines fewer** than base w/ heatmap.2
- **25 lines fewer** than pure ggplot2

The Berger, Turner, Zwick heatmap

Let's call the program from Stata this time

```
heatmap y3_trim fthomebuyers_filingunits_2000 mdate ///
        [aw=totalhsales_base], n(100) id(zip) tperiod(yearmon) ///
        splity(10) polbreak(Jan 2009, Dec 2009, Jul 2010) ///
        save(BTZRep.pdf)
```

- Default group function is mean, but the quantiles are weighted

The Berger, Turner, Zwick heatmap

Let's call the program from Stata this time

```
heatmap y3_trim fthomebuyers_filingunits_2000 mdate ///
        [aw=totalhsales_base], n(100) id(zip) tperiod(yearmon) ///
        splity(10) polbreak(Jan 2009, Dec 2009, Jul 2010) ///
        save(BTZRep.pdf)
```

- Default group function is mean, but the quantiles are weighted
- Each column is a month, labelled appropriately

The Berger, Turner, Zwick heatmap

Let's call the program from Stata this time

```
heatmap y3_trim fthomebuyers_filingunits_2000 mdate ///
        [aw=totalhsales_base], n(100) id(zip) tperiod(yearmon) ///
        splity(10) polbreak(Jan 2009, Dec 2009, Jul 2010) ///
        save(BTZRep.pdf)
```

- Default group function is mean, but the quantiles are weighted
- Each column is a month, labelled appropriately
- `polbreak()` interprets time strings and adds policy lines accordingly

The Berger, Turner, Zwick heatmap

Let's call the program from Stata this time

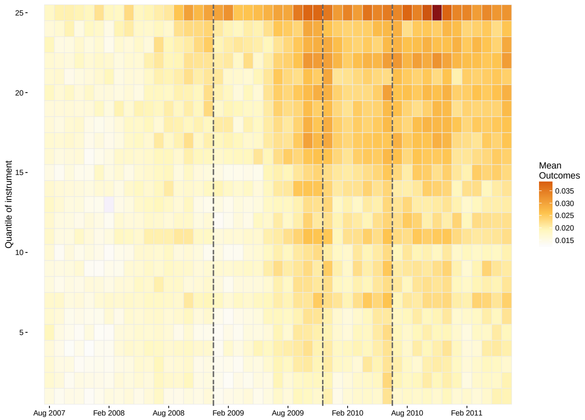
```
heatmap y3_trim fthomebuyers_filingunits_2000 mdate ///
        [aw=totalhsales_base], n(100) id(zip) tperiod(yearmon) ///
        splity(10) polbreak(Jan 2009, Dec 2009, Jul 2010) ///
        save(BTZRep.pdf)
```

- Default group function is mean, but the quantiles are weighted
- Each column is a month, labelled appropriately
- `polbreak()` interprets time strings and adds policy lines accordingly
- `splity(n)` divides y-axis labels into n even intervals

The Berger, Turner, Zwick heatmap

Another perspective: check the standard errors on the mean estimates over a coarser partition

```
heatmap y3_trim fthomebuyers_filingunits_2000 mdate ///
       [aw=totalhsales_base], n(25) id(zip) tperiod(yearmon) ///
        grpfunc(sem) splity(5) count out ///
        polbreak(Jan 2009, Dec 2009, Jul 2010) save(BTZRep_se.pdf)
```



Conclusions

When not to use heatmaps

- Heatmaps are not a panacea: there is a tradeoff between
  - The additional information they effectively display;
  - The information lost in using colours to represent change instead of geometric shapes
- It is also unclear how heatmaps can display uncertainty of estimates: distribution of estimates, e.g.?
- A good argument for a package that simplifies heatmap creation — the less time spent making a visualization, the less likely one gets overattached to one when a better solution exists

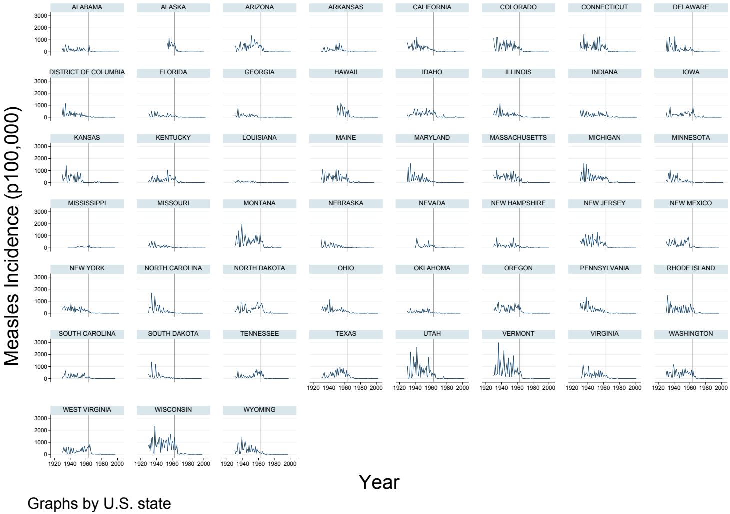
When not to use heatmaps

A good heuristic (define Z as the variable plotted with colour):

- Plotting quantiles on the Y axis: Is your graph confounded if you plotted Z against X in overlapping line graphs split by Y?
- Plotting a factor variable on the Y axis: Is your graph confounded if you plotted Z against X in a small multiples plot split by Y?

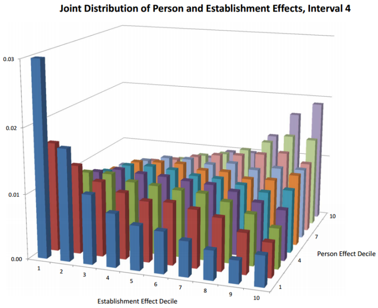
When not to use heatmaps

Example: Measles vaccine revisited



When not to use heatmaps

Example: visualizing positive assortative matching



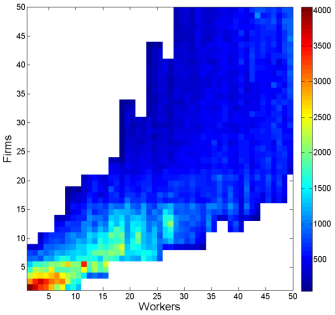


Figure 6: Estimated Match Density.

(L: Card, Heining & Kline (2012); R: Hagedorn, Law & Manovskii (2016)) 2016 How would the interpretation change if the visualization was instead overlaying many marginals over each other? Small multiples of marginals?

Future updates

- Syntax revisions
- Complementary side plots (histograms, time series, diffs. . .)
- Both axes can belong in one of four types
- Port the heatmap palette for utilisation in base R heatmap f'n
- ???

References

Thanks!