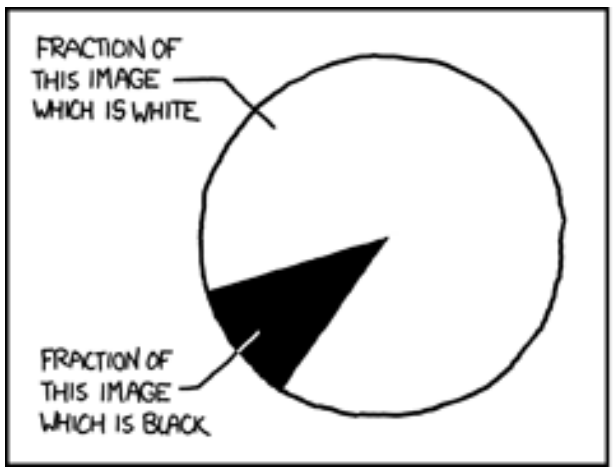
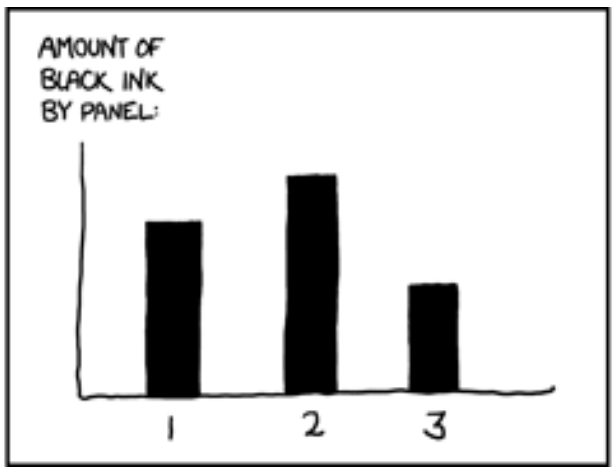
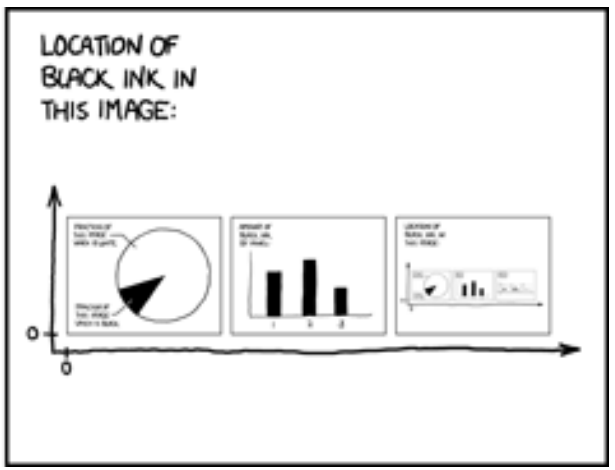
Data Visualization

Dennis Wylie, UT Bioinformatics Consulting Group

April 11, 2017


FRACTION OF THIS IMAGE WHICH IS WHITE
FRACTION OF THIS IMAGE WHICH IS BLACK


AMOUNT OF BLACK INK BY PANEL:
1
2
3


LOCATION OF BLACK INK IN THIS IMAGE:
0
0

Types of visualization

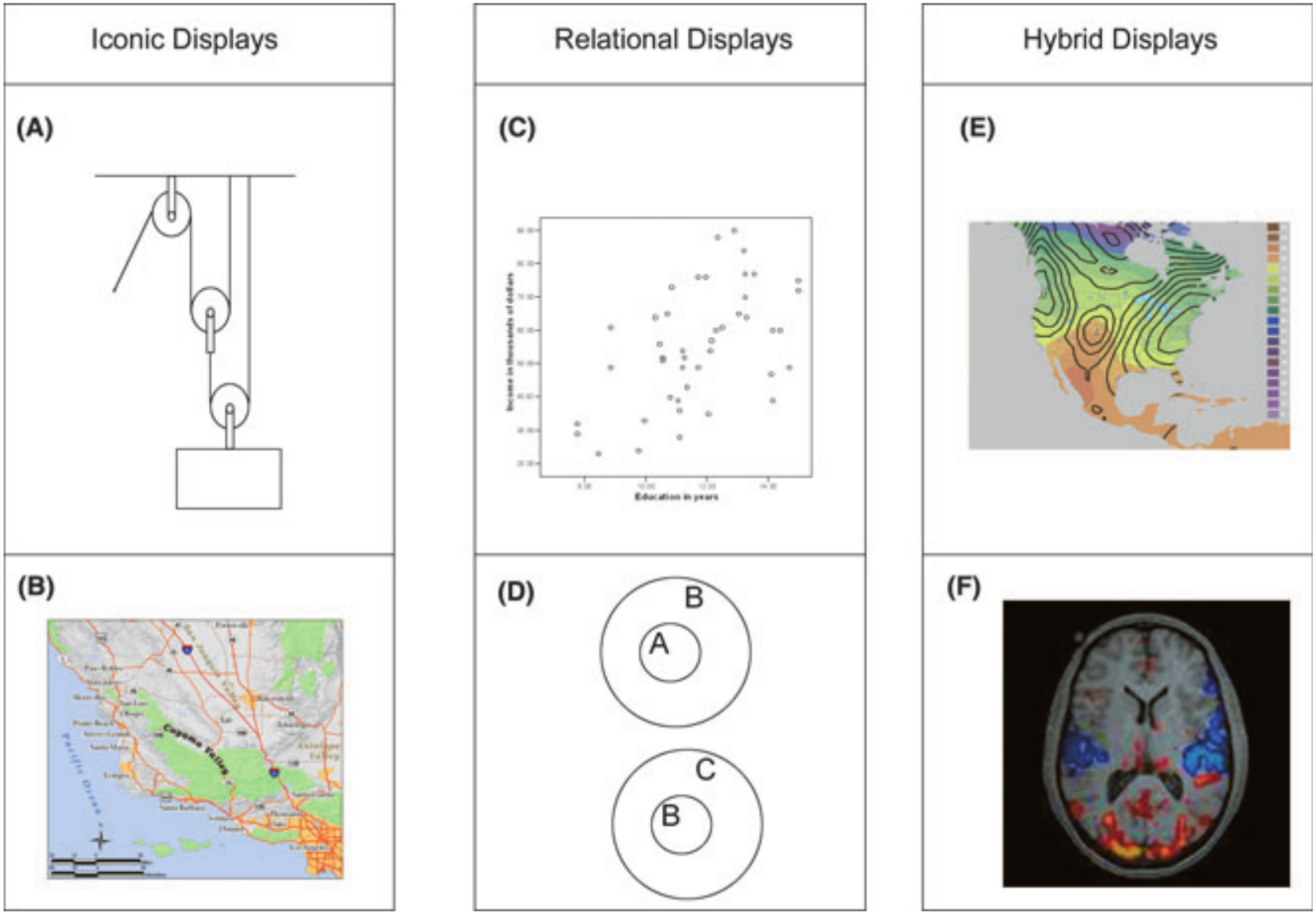


Fig. 1. Examples of iconic, relational, and hybrid displays.

History of data visualization

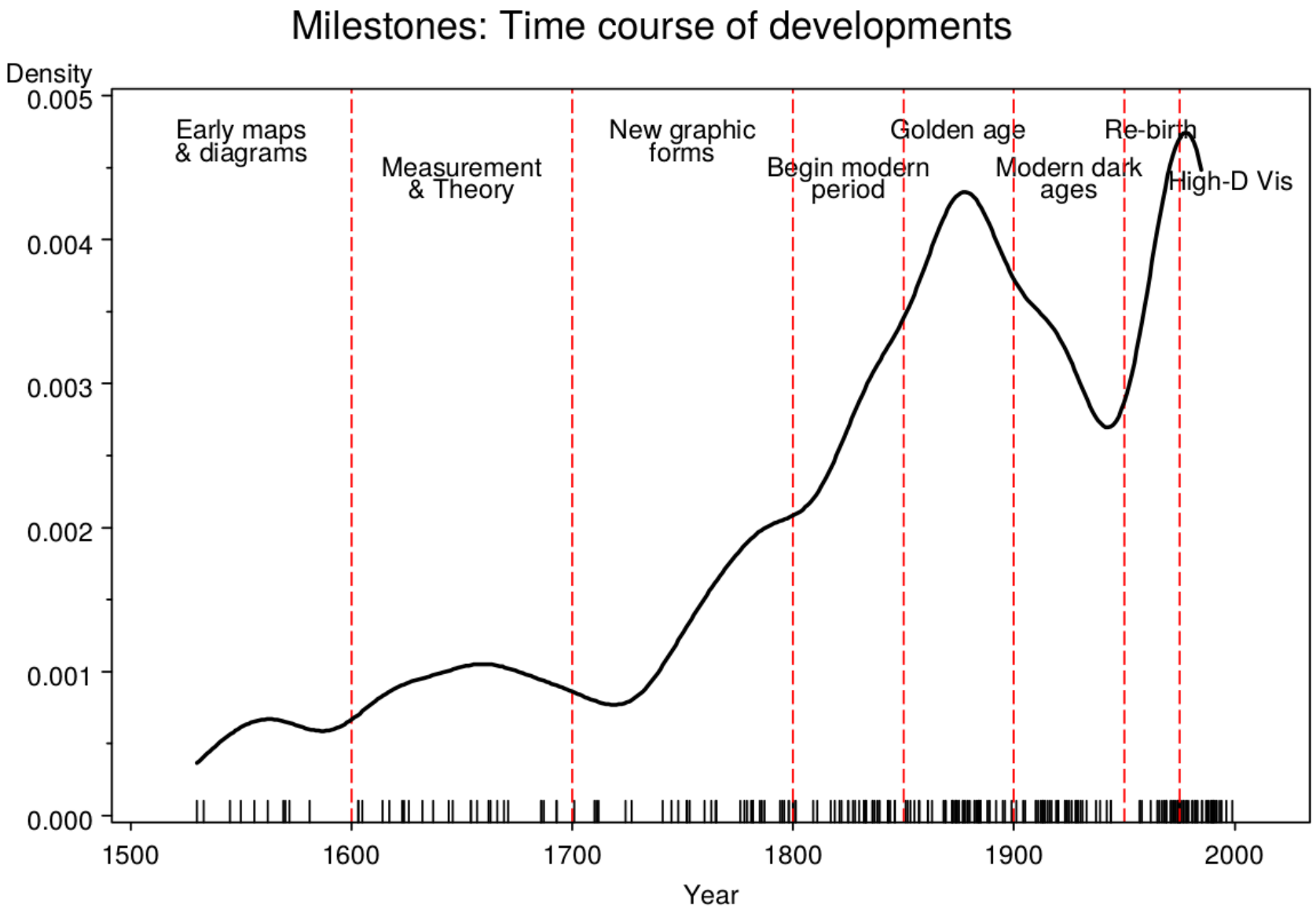
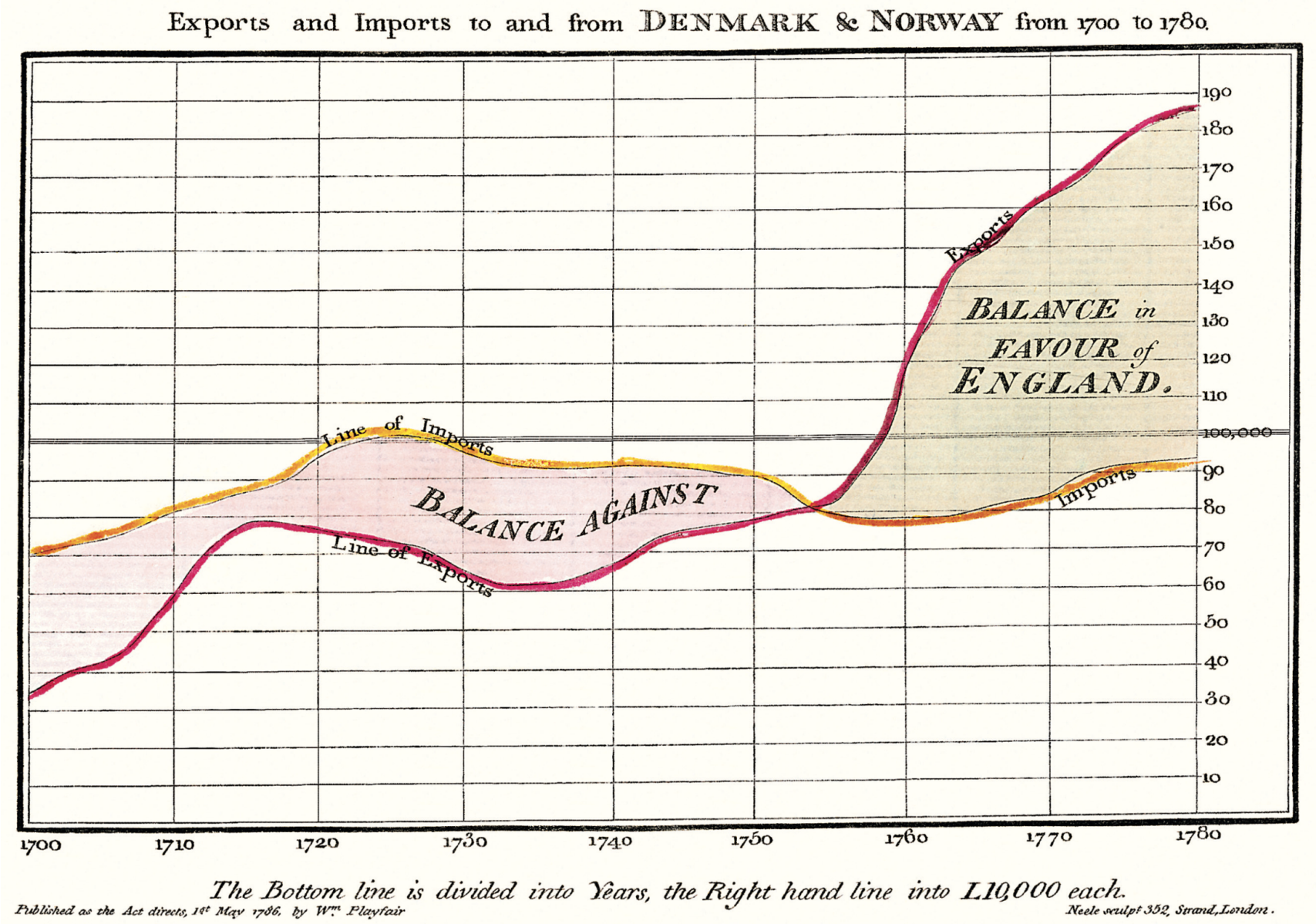
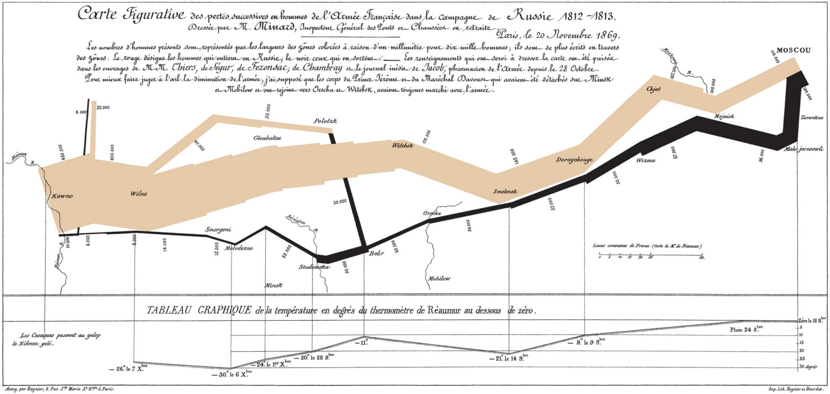


Figure 1: The time distribution of events considered milestones in the history of data visualization, shown by a rug plot and density estimate.

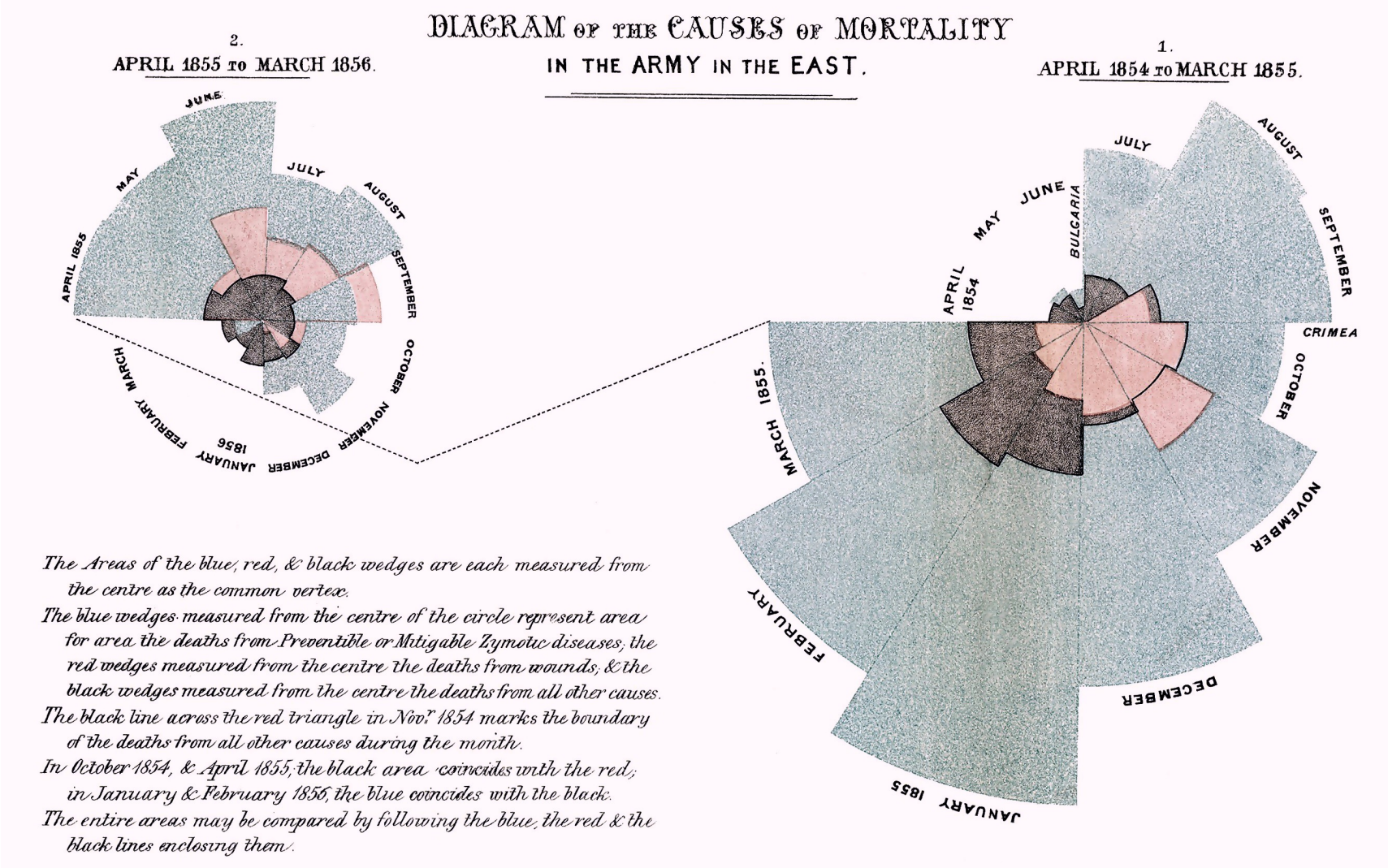
Playfair



Minard


Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Kowno
Wilna
Smorgoni
Molodezno
Gloubokoe
Polotzk
Witebsk
Smolensk
Dorogobouge
Wizma
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Orscha
Bobr
Studienska
Minsk
Mohilow
Niémen
422.000
400.000
22.000
6.000
60.000
33.000
30.000
100.000
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
— 26° le 7 X.bre
— 30° le 6 X.bre
— 24° le 1er X.bre
— 20° le 28 9.bre
— 11°
— 21° le 14 9.bre
— 9° le 9 9.bre
Pluie 24 8.bre
Zéro le 18 8.bre
30 degrés

Nightingale



Why visualize?

- Summarize data

Why visualize?

- Summarize data
- Illustrate relationships

Why visualize?

- Summarize data
- Illustrate relationships
- Pattern recognition by human visual perception

Why visualize?

- Summarize data
- Illustrate relationships
- Pattern recognition by human visual perception
  - map properties to "aesthetics"

Why visualize?

- Summarize data
- Illustrate relationships
- Pattern recognition by human visual perception
  - map properties to "aesthetics"

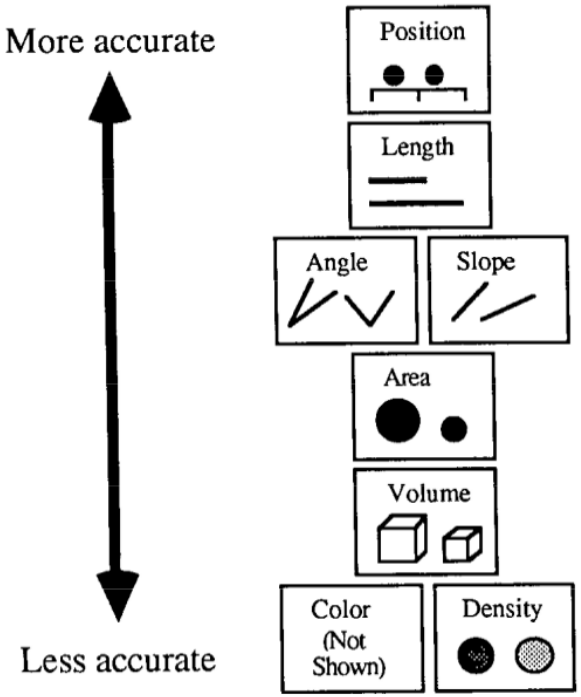
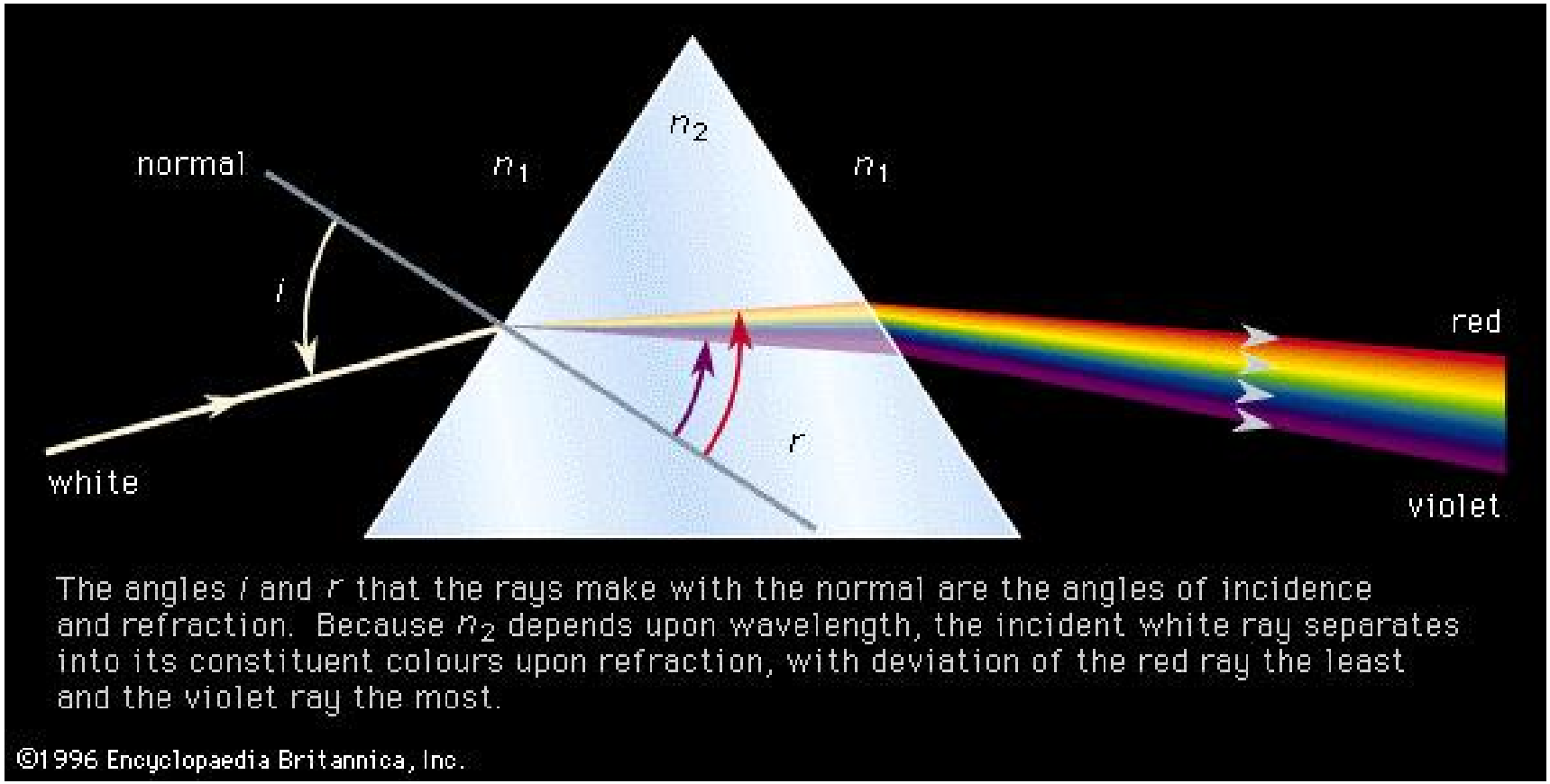
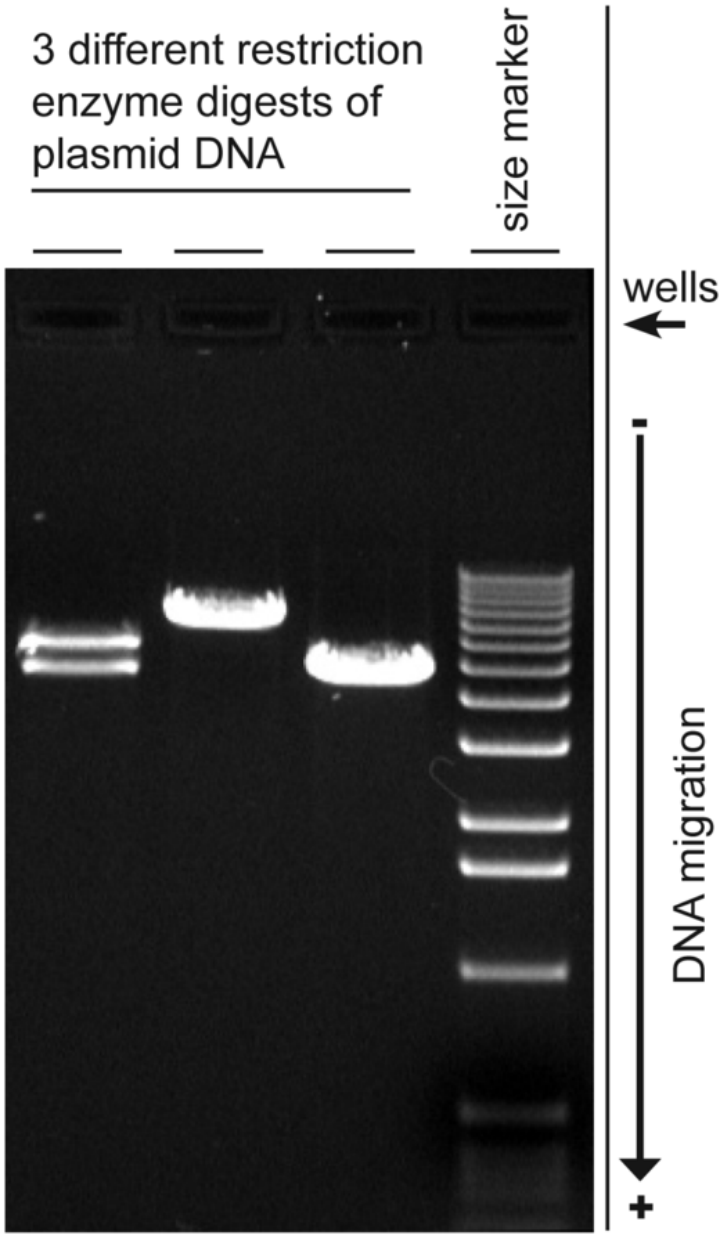


Fig. 14. Accuracy ranking of quantitative perceptual tasks. Higher tasks are accomplished more accurately than lower tasks. Cleveland and McGill empirically verified the basic properties of this ranking.

Natural example: frequency mapped to angle



Experimental example: size mapped to position



- Discrete **experiment/condition** mapped to horizontal position
- Continuous(ish) **molecular size** mapped to vertical position
- Continuous(ish) **copy number** mapped to brightness

Choosing the right aesthetic

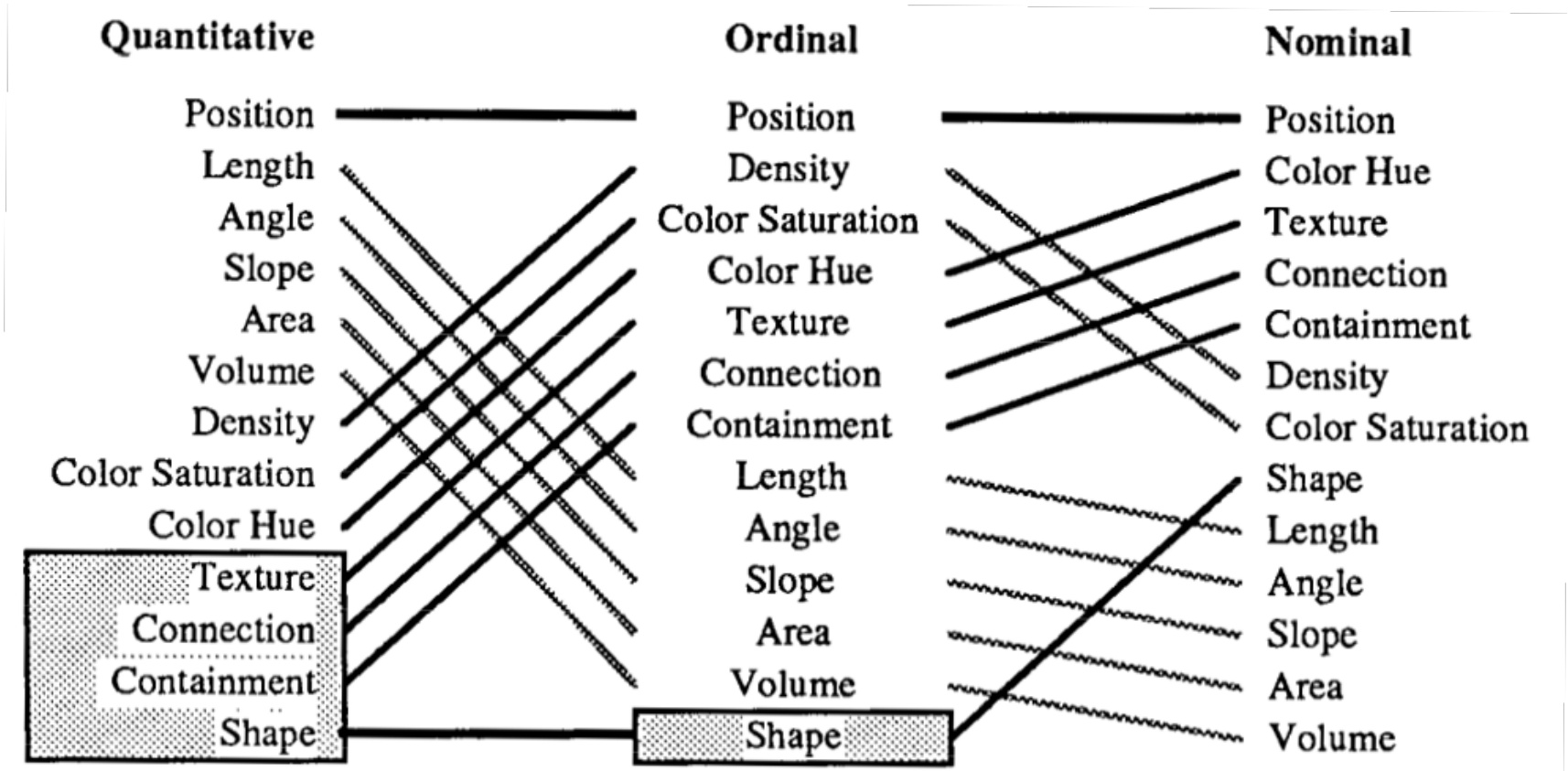
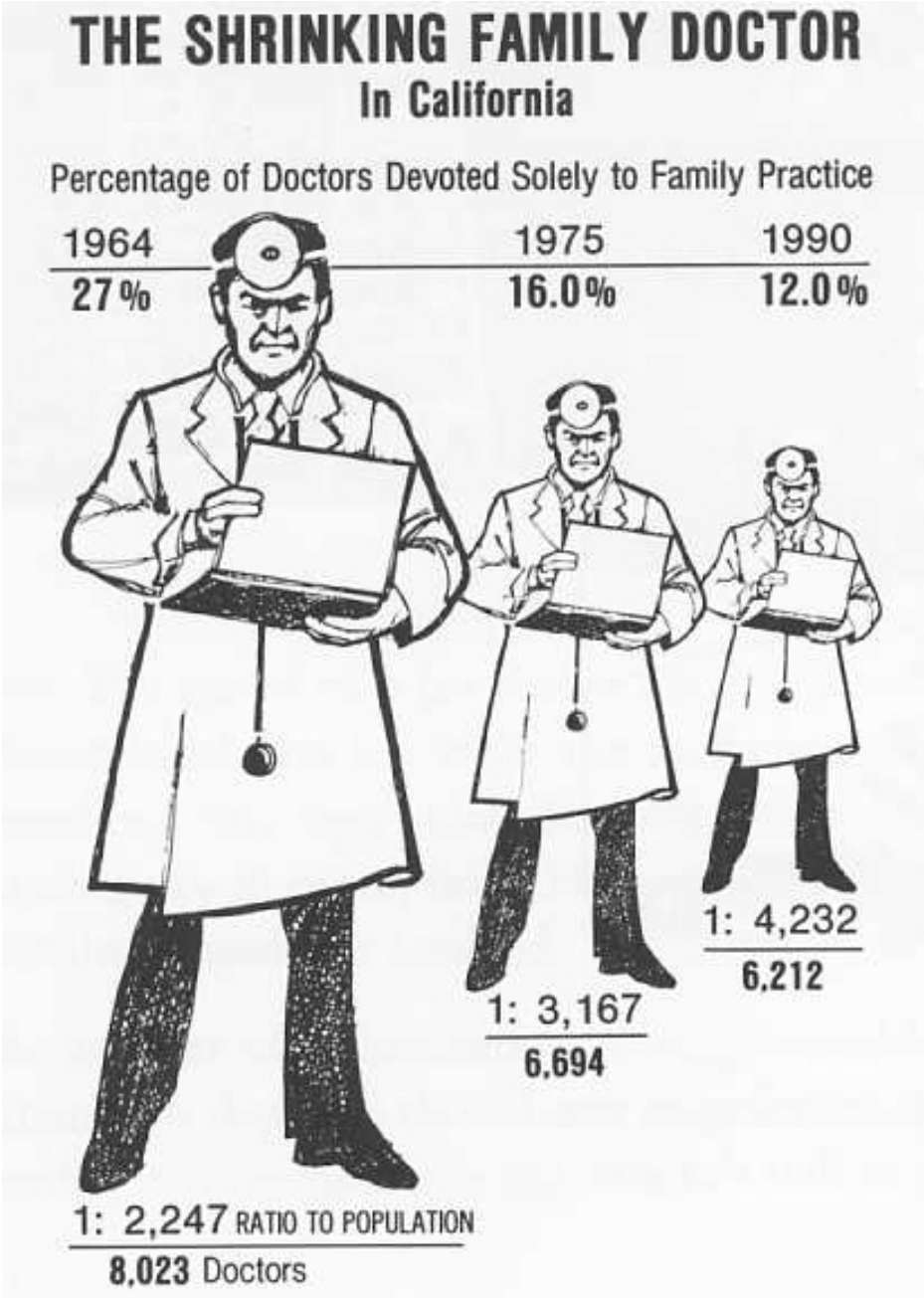
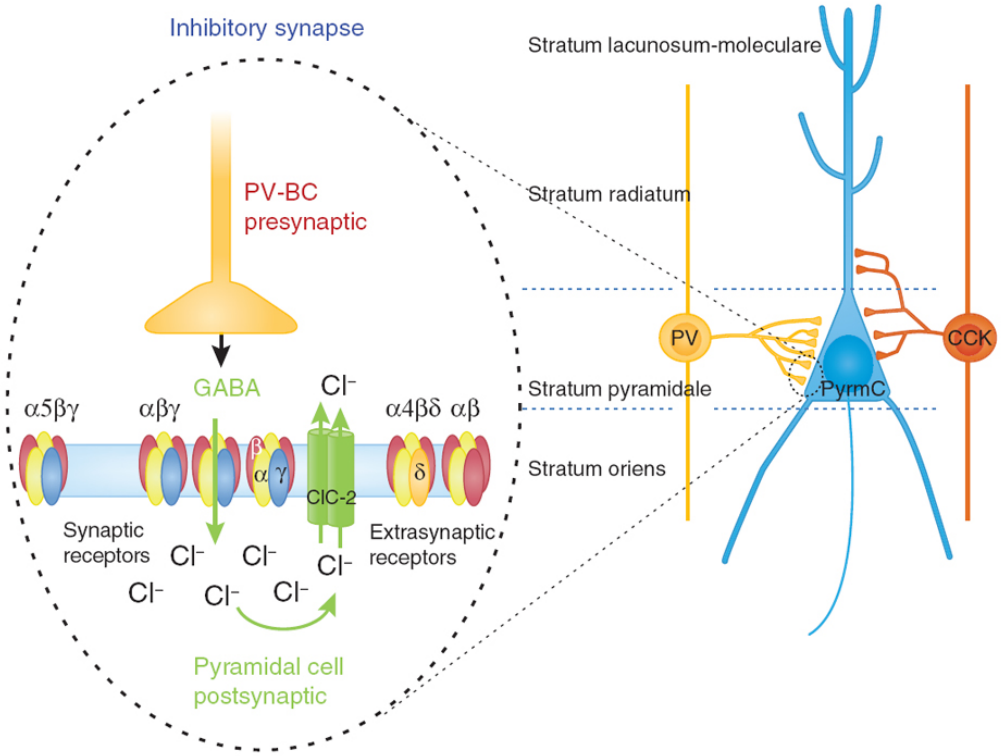


Fig. 15. Ranking of perceptual tasks. The tasks shown in the gray boxes are not relevant to these types of data.

“Lie factor” from improper area encoding



Neuron subtypes and GSE75386

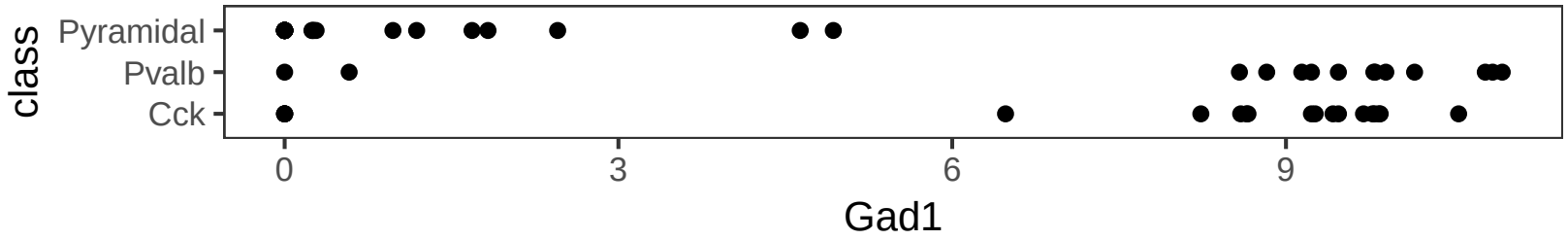


GSE75386 Single cell RNAseq of electrophysiologically characterized neurons of the hippocampus

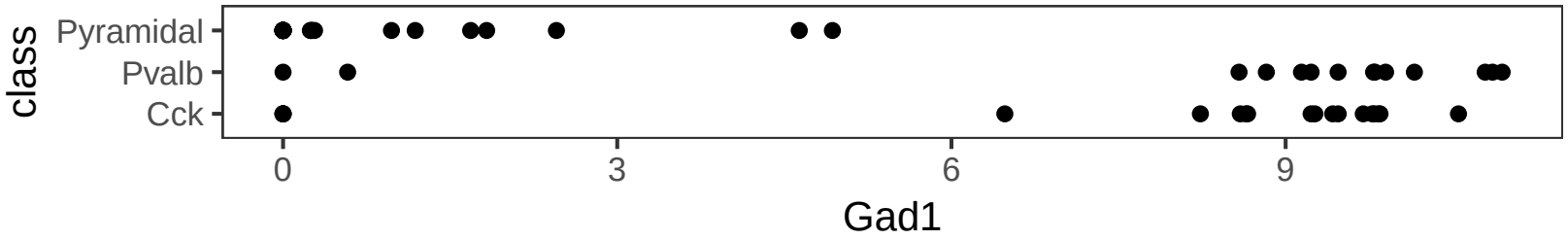
Organism Mus musculus

Design 15 tissue samples + 93 single-cell samples, including 58 Cck, Pvalb, and Pyramidal neurons.

GSE75386 stripchart example

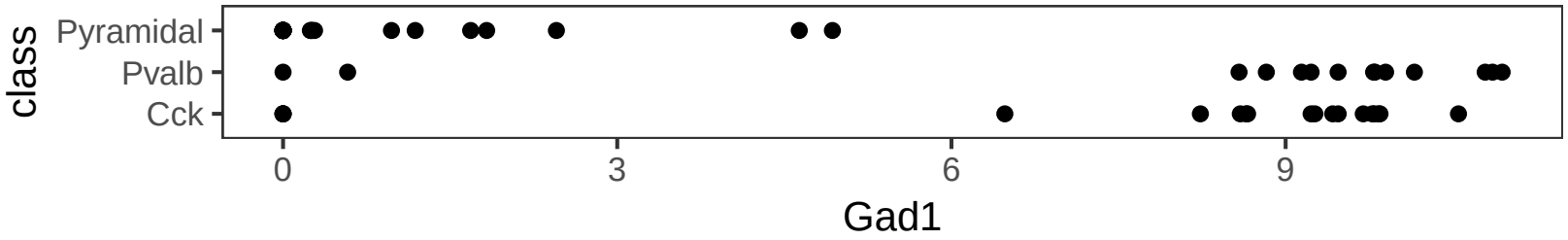


GSE75386 stripchart example




```
ggstrip = ggplot(
    data = gse75386,
    mapping = aes(
        x = Gad1,
        y = class
    )
)
ggstrip = ggstrip + geom_point()
print(ggstrip)
```

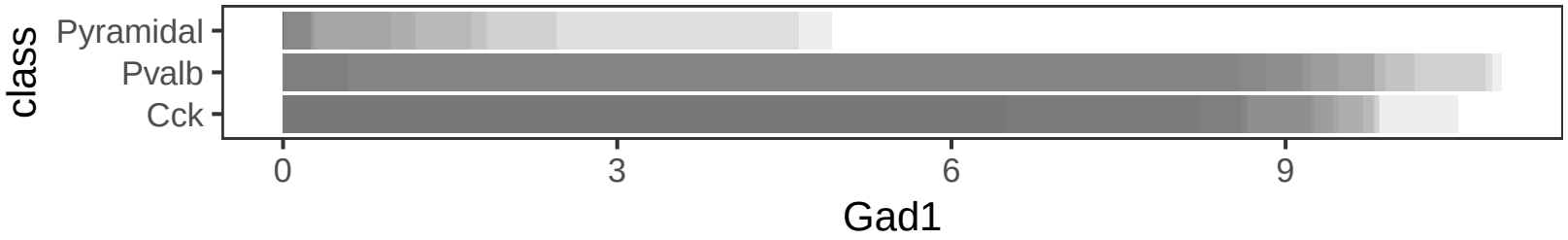
GSE75386 stripchart example




```
ggstrip = ggplot(
    data = gse75386,
    mapping = aes(
        x = Gad1,
        y = class
    )
)
ggstrip = ggstrip + geom_point()
print(ggstrip)
```

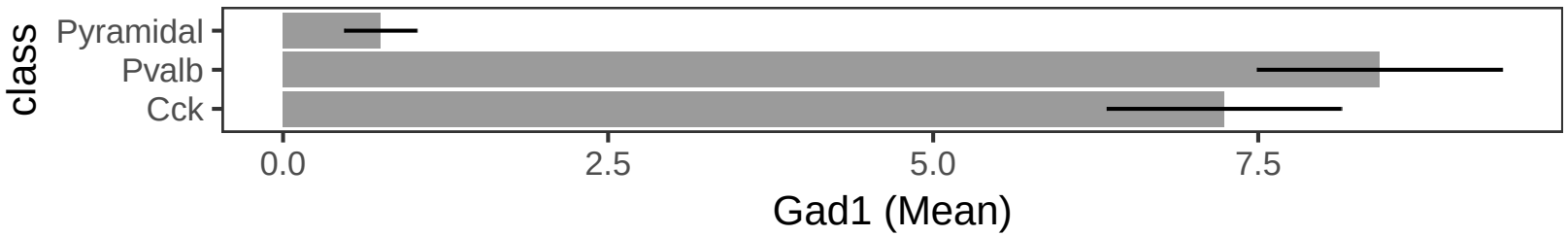
What is "geom_point"?

GSE75386 overplotted bars




```
ggbar = ggplot(gse75386, aes(x=class, y=Gad1))
ggbar = ggbar + geom_bar(alpha=0.1,
                         position='identity', stat='identity')
ggbar = ggbar + coord_flip()
print(ggbar)
```

GSE75386 mean bars + SE lines

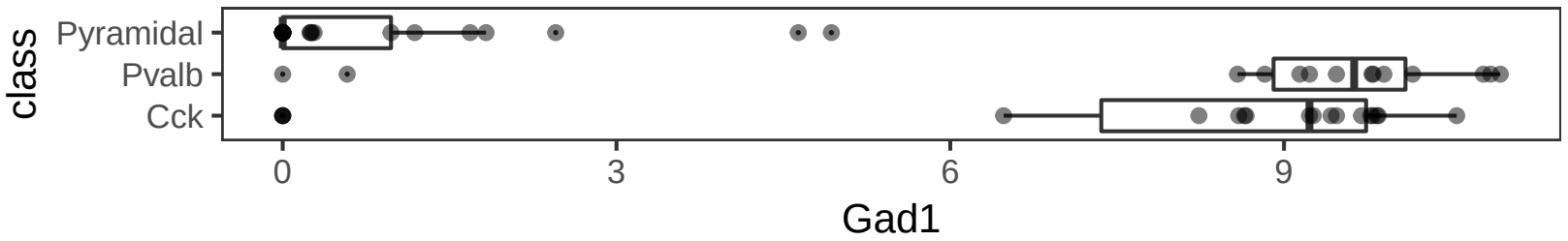



```
gse75386stats = gse75386 %>%
                group_by(class) %>%
                summarize(
                    `Gad1 (Mean)` = mean(Gad1),
                    SE = sd(Gad1) / sqrt(length(Gad1))
                )
ggbarse = ggplot(gse75386stats, aes(x=class, y=`Gad1 (Mean)`))
ggbarse = ggbarse + geom_bar(alpha=0.6, stat='identity')
ggbarse = ggbarse + geom_errorbar(aes(ymin=`Gad1 (Mean)` - SE,
                                      ymax=`Gad1 (Mean)` + SE),
                                  width=0)
ggbarse = ggbarse + coord_flip()
print(ggbarse)
```

Layered grammar of graphics

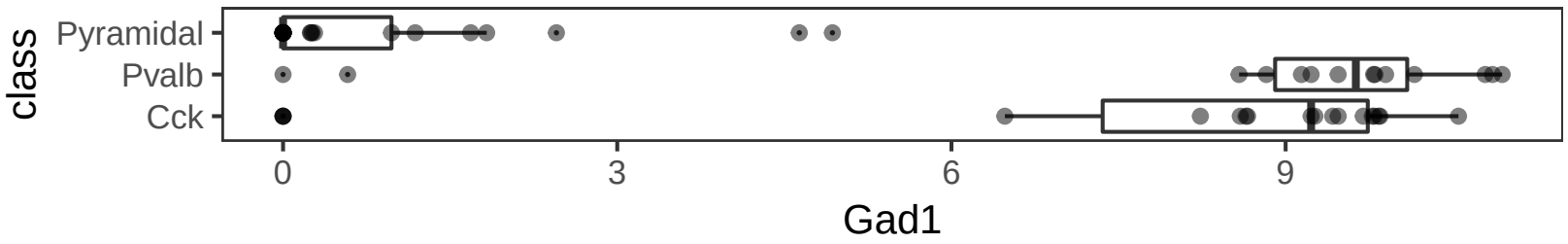
$$\begin{aligned} \text{layer} = \ & \text{data} + \\ & \textbf{stat}\text{istical transformation} + \\ & \textbf{aes}\text{thetic} + \\ & \textbf{geom}\text{etric object} \end{aligned}$$

GSE75386 boxplot + stripchart




```
ggbox = ggplot(gse75386, aes(x=class, y=Gad1))
ggbox = ggbox + geom_boxplot(stat='boxplot',
                             outlier.size=0)
ggbox = ggbox + geom_point(alpha=0.5)
ggbox = ggbox + coord_flip()
```

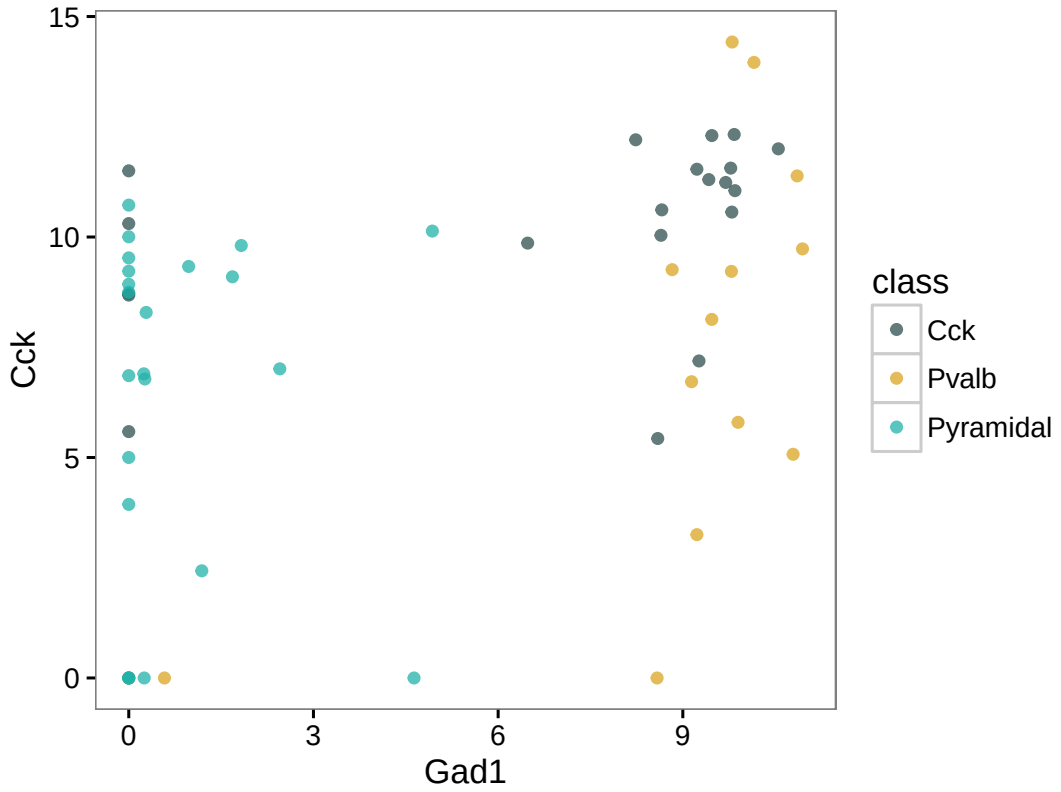
GSE75386 boxplot + stripchart




```
ggbox = ggplot(gse75386, aes(x=class, y=Gad1))
ggbox = ggbox + geom_boxplot(stat='boxplot',
                             outlier.size=0)
ggbox = ggbox + geom_point(alpha=0.5)
ggbox = ggbox + coord_flip()
```

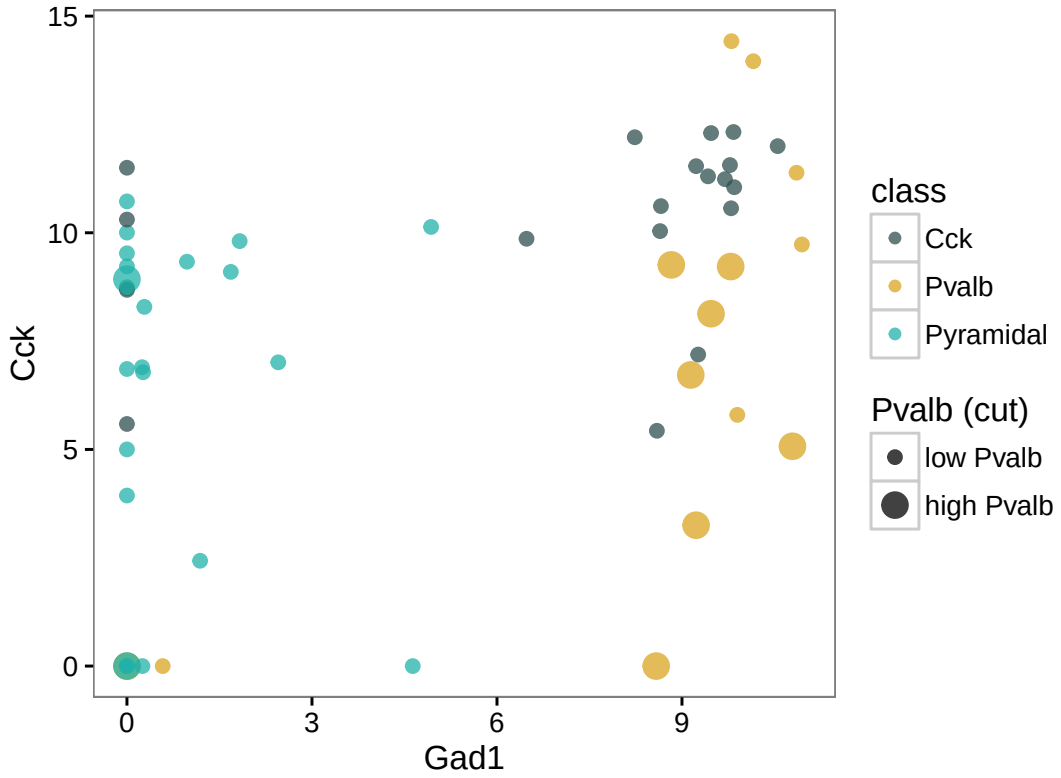
. . . some statistical transformations are built in to `ggplot`.

GSE75386 scatterplot



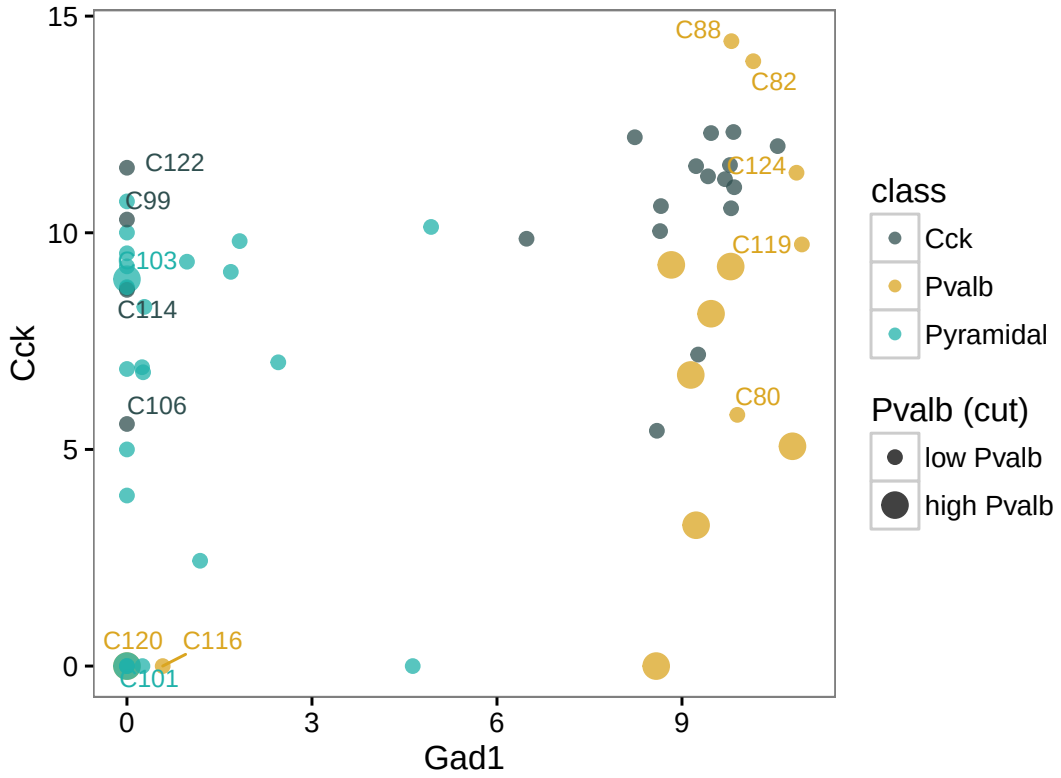

```
ggscat = ggplot(
    gse75386,
    aes(x=Gad1, y=Cck, color=class)
)
ggscat = ggscat + geom_point(alpha=0.75)
ggscat = ggscat + scale_color_manual(
        values=c('darkslategray', 'goldenrod', 'lightseagreen'))
```

GSE75386 scatterplot



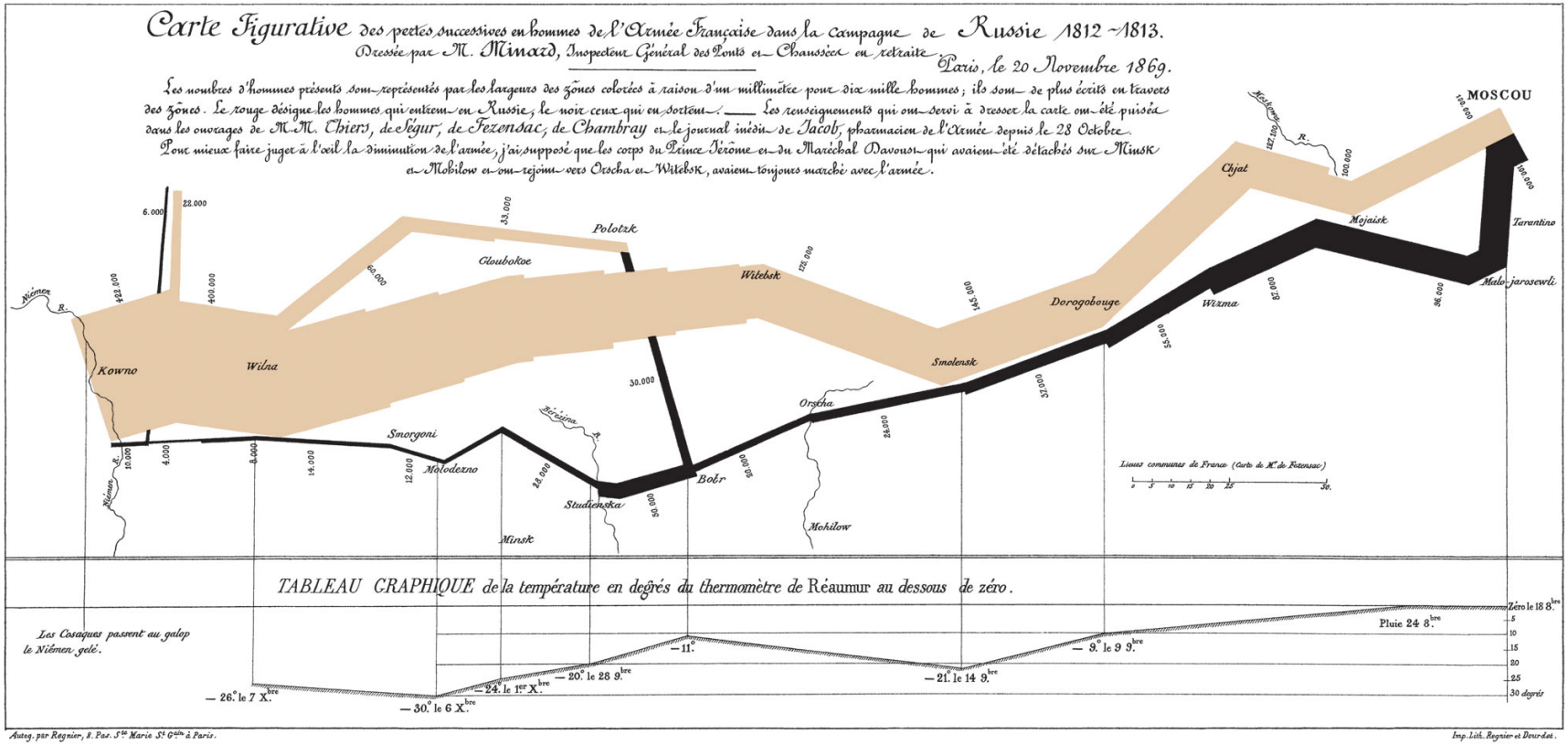

```
ggscat = ggplot(
    gse75386,
    aes(x=Gad1, y=Cck, color=class, size=`Pvalb (cut)`)
)
ggscat = ggscat + geom_point(alpha=0.75)
ggscat = ggscat + scale_color_manual(
        values=c('darkslategray', 'goldenrod', 'lightseagreen'))
ggscat = ggscat + scale_size_manual(values=c(2, 4))
```

GSE75386 scatterplot + text layer

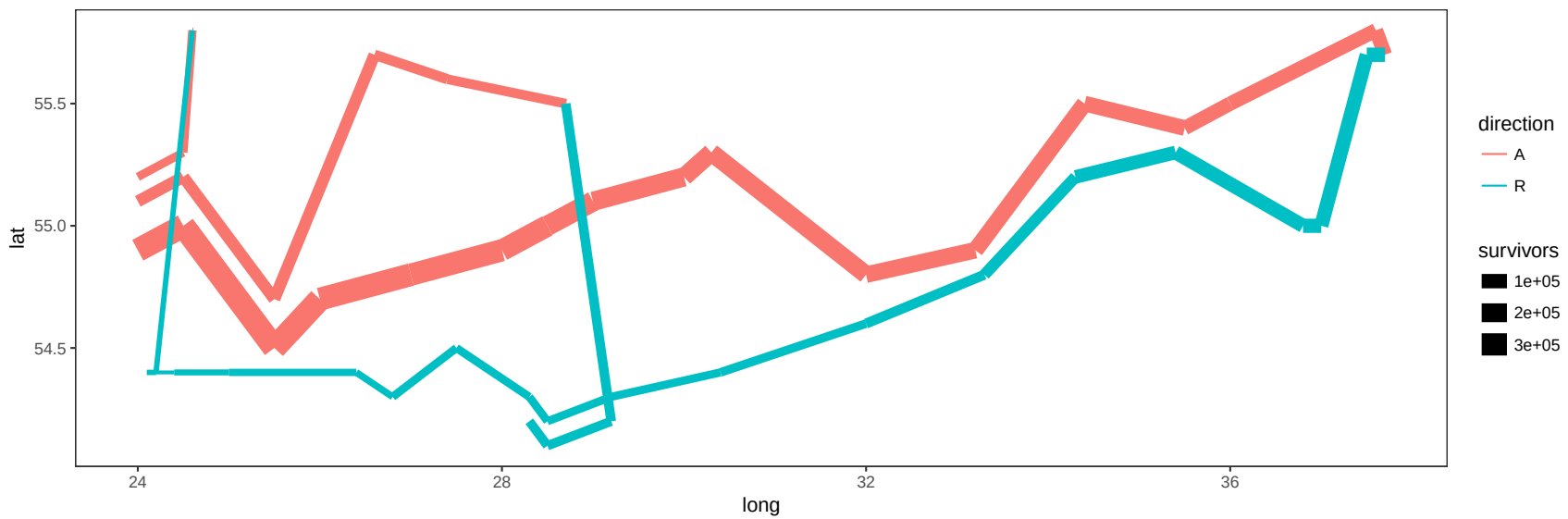



```
...
ggscat = ggscat + geom_text_repel(
    aes(label=odd),
    size = 3,
    show.legend = FALSE
)
```

The best graph ever?

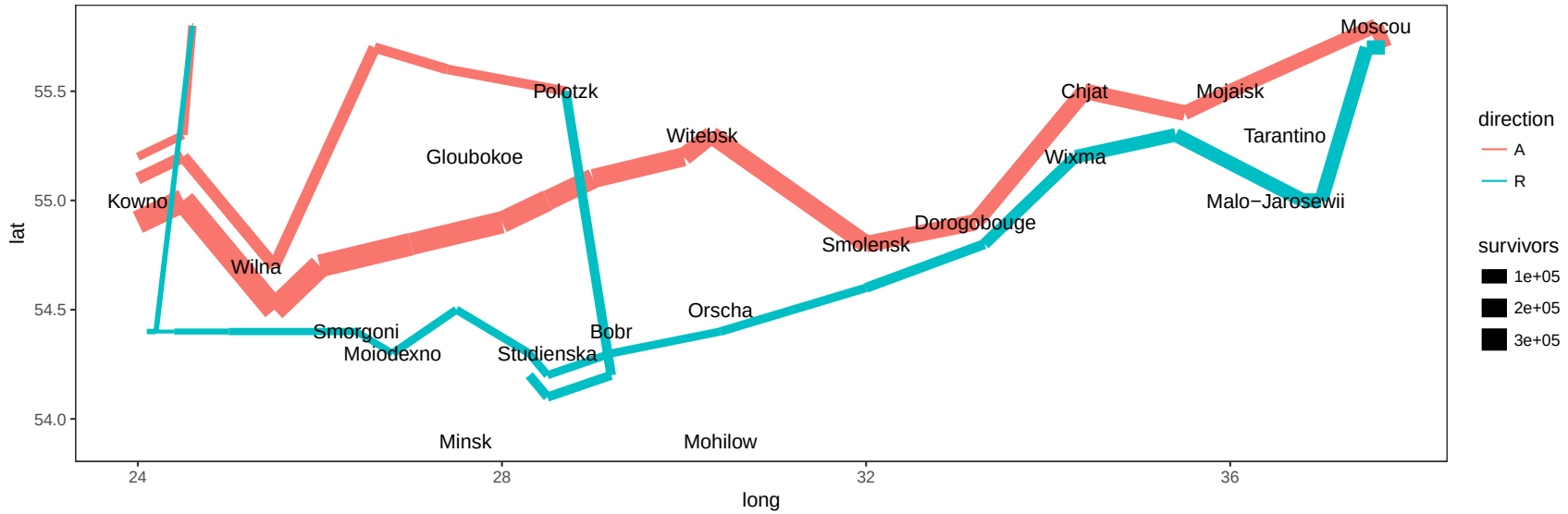


ggplotting Minard



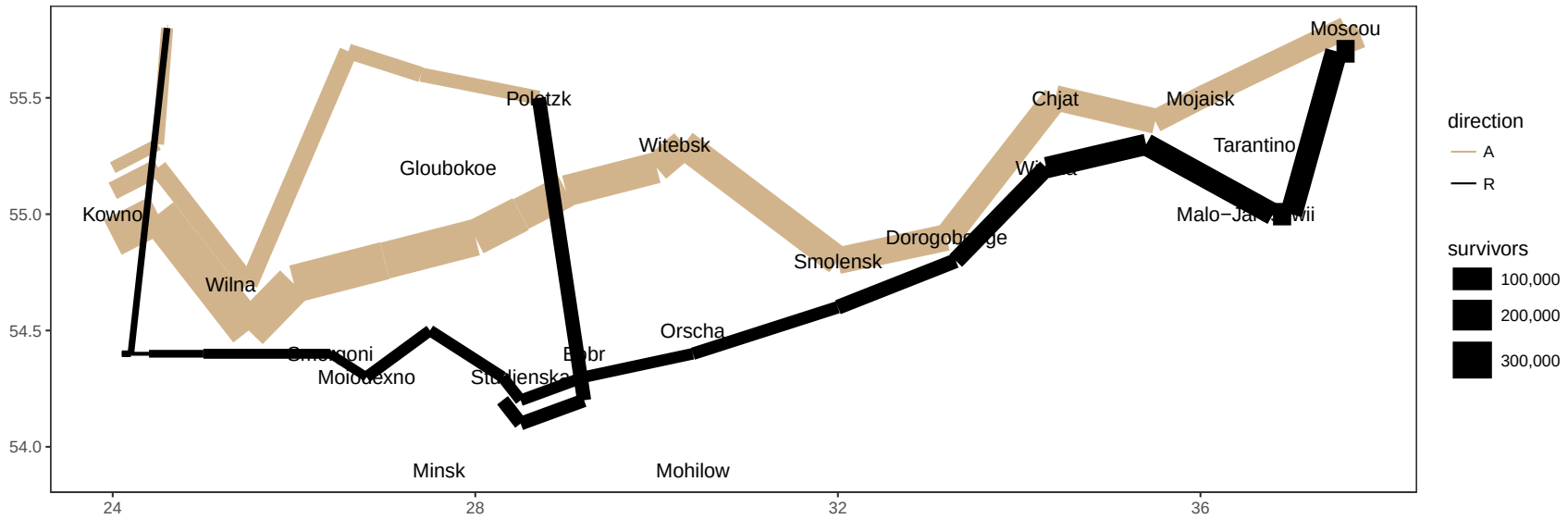

```
ggtroops = ggplot(troops, aes(long, lat))
ggtroops = ggtroops + geom_path(aes(
    size = survivors,
    color = direction,
    group = group
))
```

ggplotting Minard



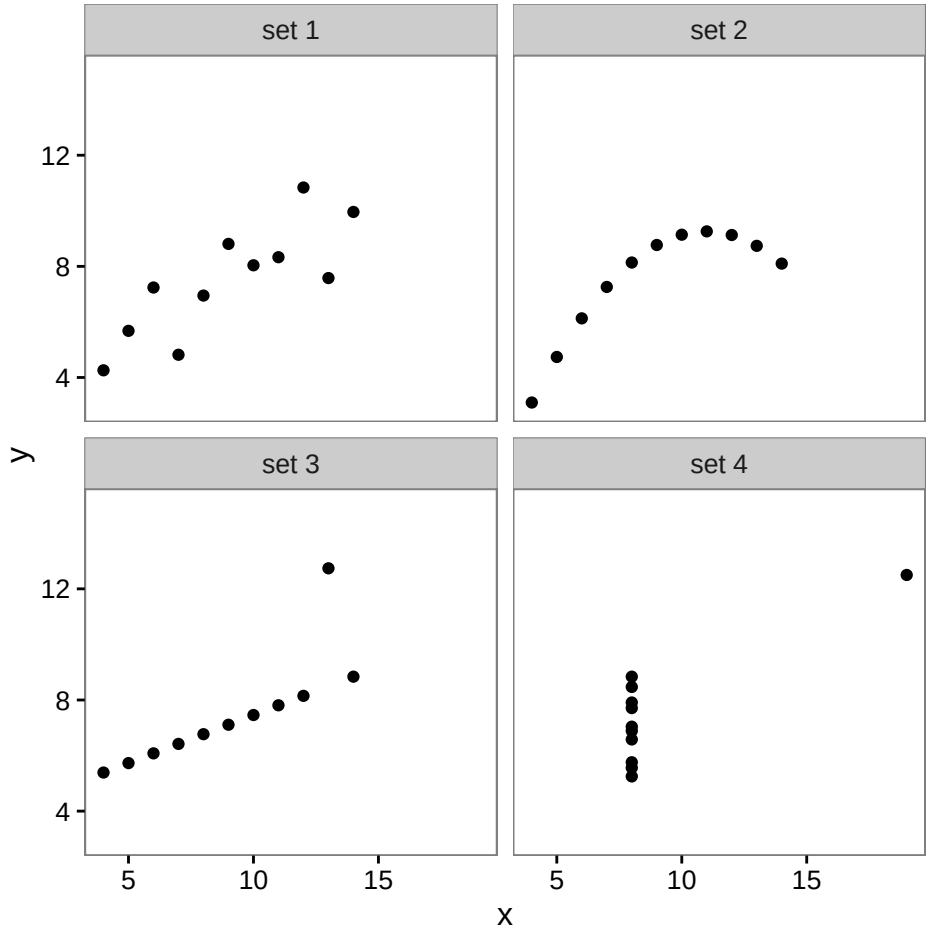

```
ggboth = ggtroops + geom_text(
    aes(label = city),
    size = 4,
    data = cities
)
```

ggplotting Minard



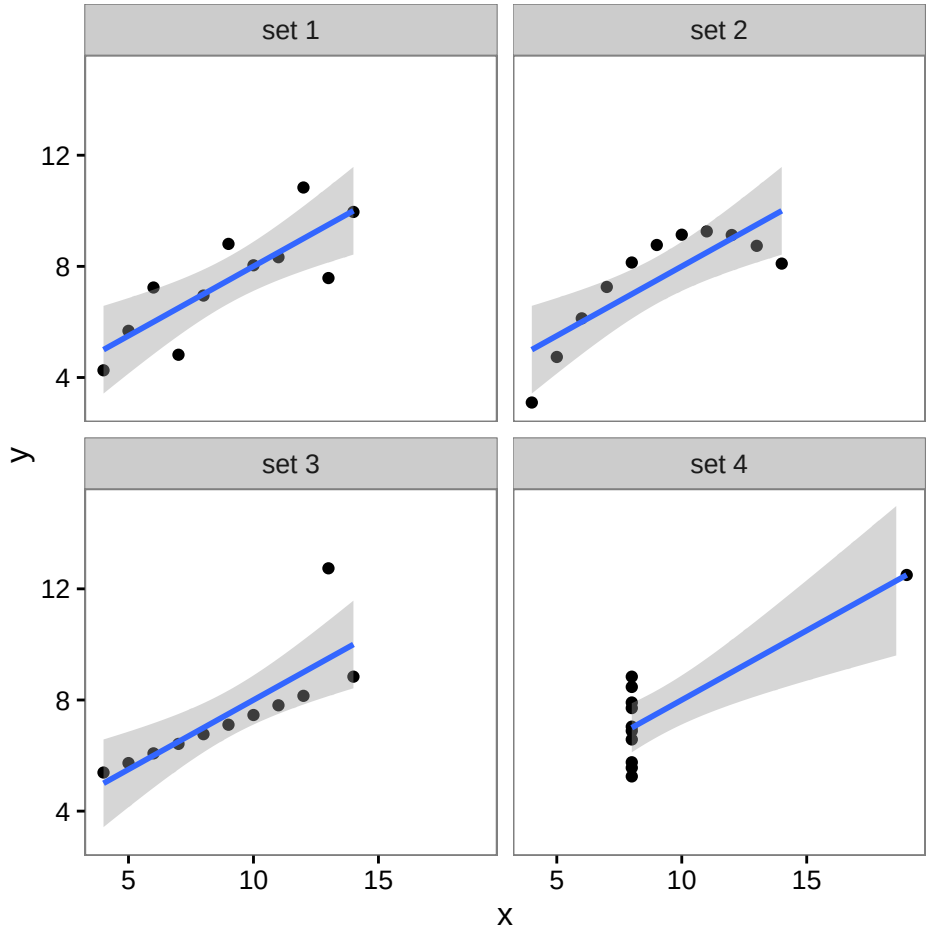

```
ggboth = ggboth + scale_size(
    range = c(1, 10),
    breaks = c(1, 2, 3) * 10^5,
    labels = comma(c(1, 2, 3) * 10^5)
)
ggboth = ggboth + scale_color_manual(values = c("#d2b48c","black"))
ggboth = ggboth + xlab(NULL) + ylab(NULL)
```

Small multiples and facetting



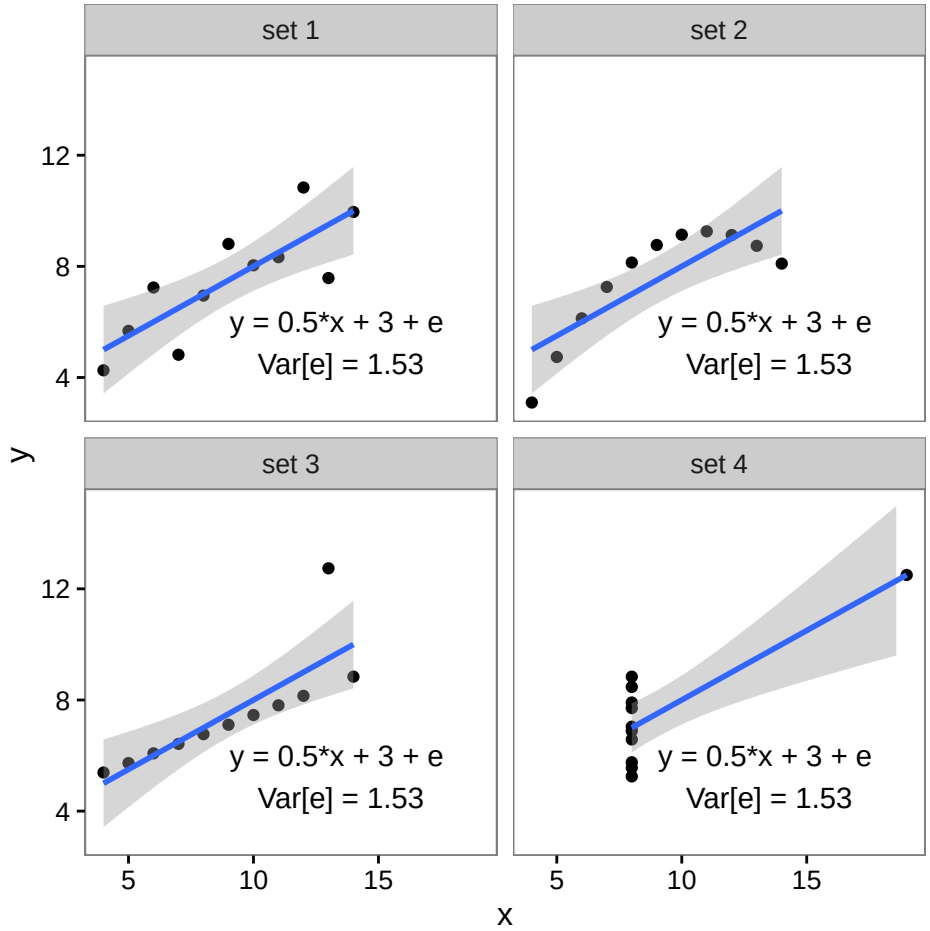

```
ggo = ggplot(anscombe, aes(x=x, y=y))
ggo = ggo + facet_wrap(~ set)
ggo = ggo + geom_point()
```

. . . with smoothed linear fit



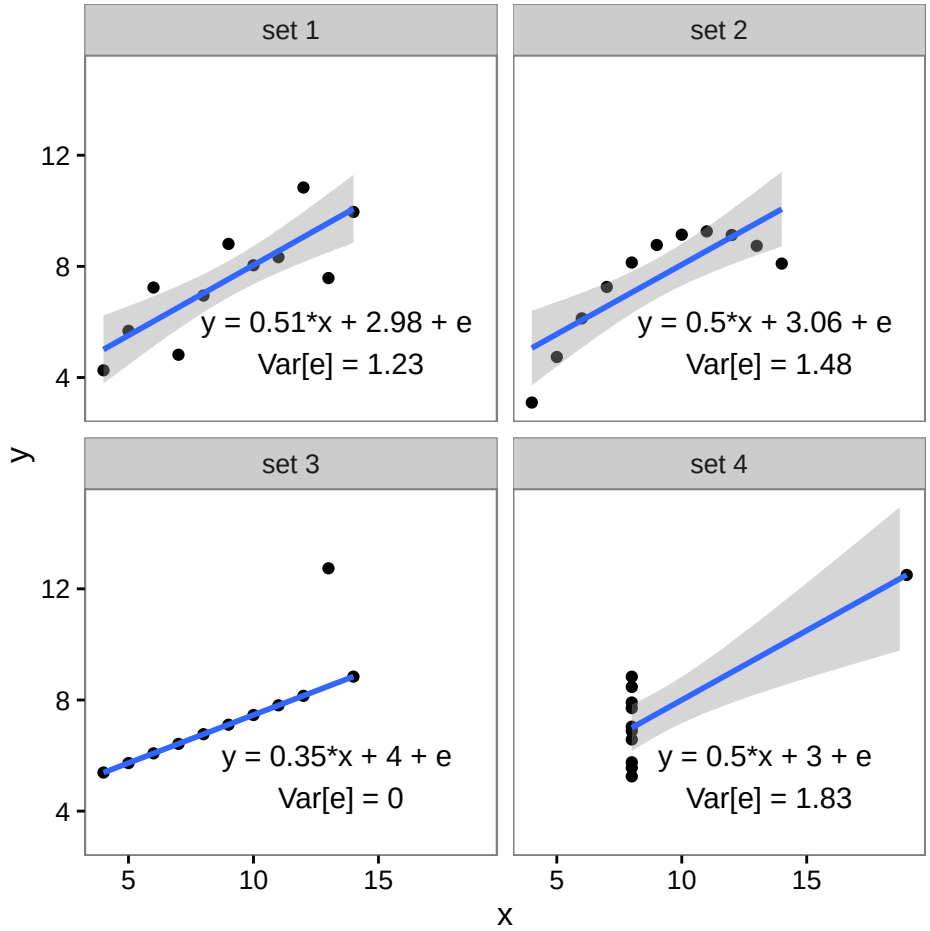

```
ggo = ggplot(anscombe, aes(x=x, y=y))
ggo = ggo + facet_wrap(~ set)
ggo = ggo + geom_point()

ggo = ggo + stat_smooth(method=lm)
```

. . . and text layer



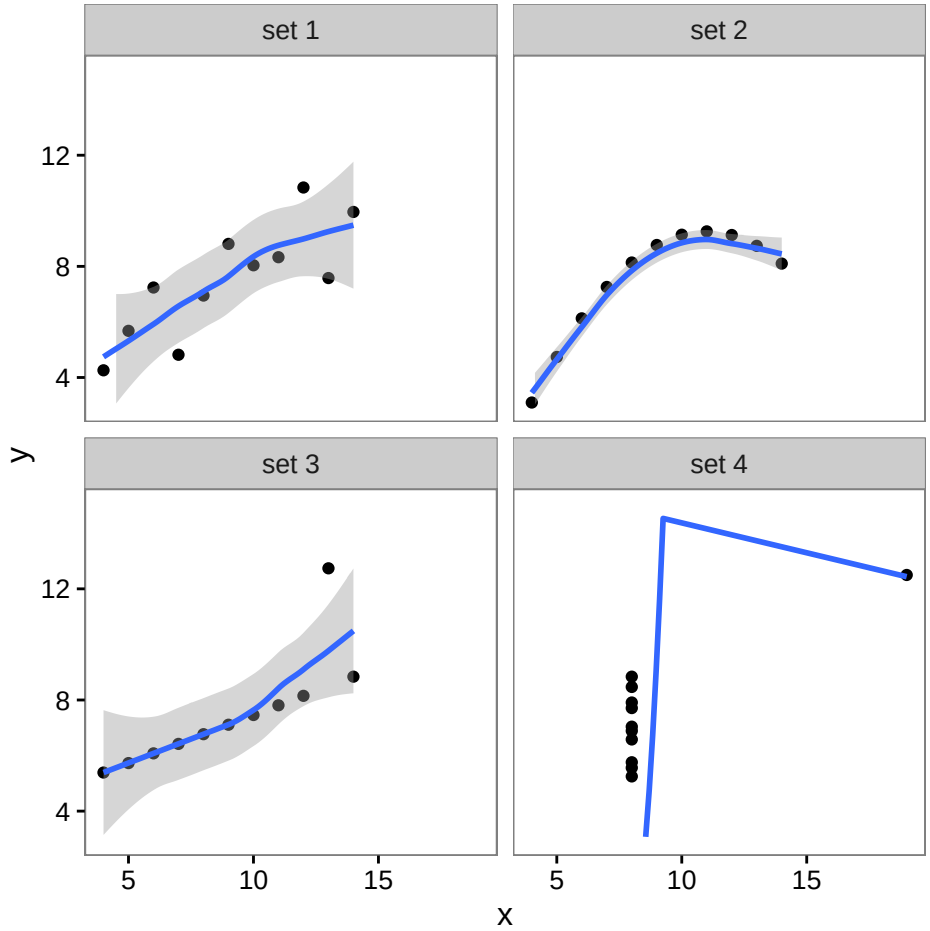

```
ggo = ggplot(anscombe, aes(x=x, y=y))
ggo = ggo + facet_wrap(~ set)
ggo = ggo + geom_point()
ggo = ggo + geom_text(aes(label=text), data=lmDescriptions(anscombe))
ggo = ggo + stat_smooth(method=lm)
```

Other smoothing methods exist as well



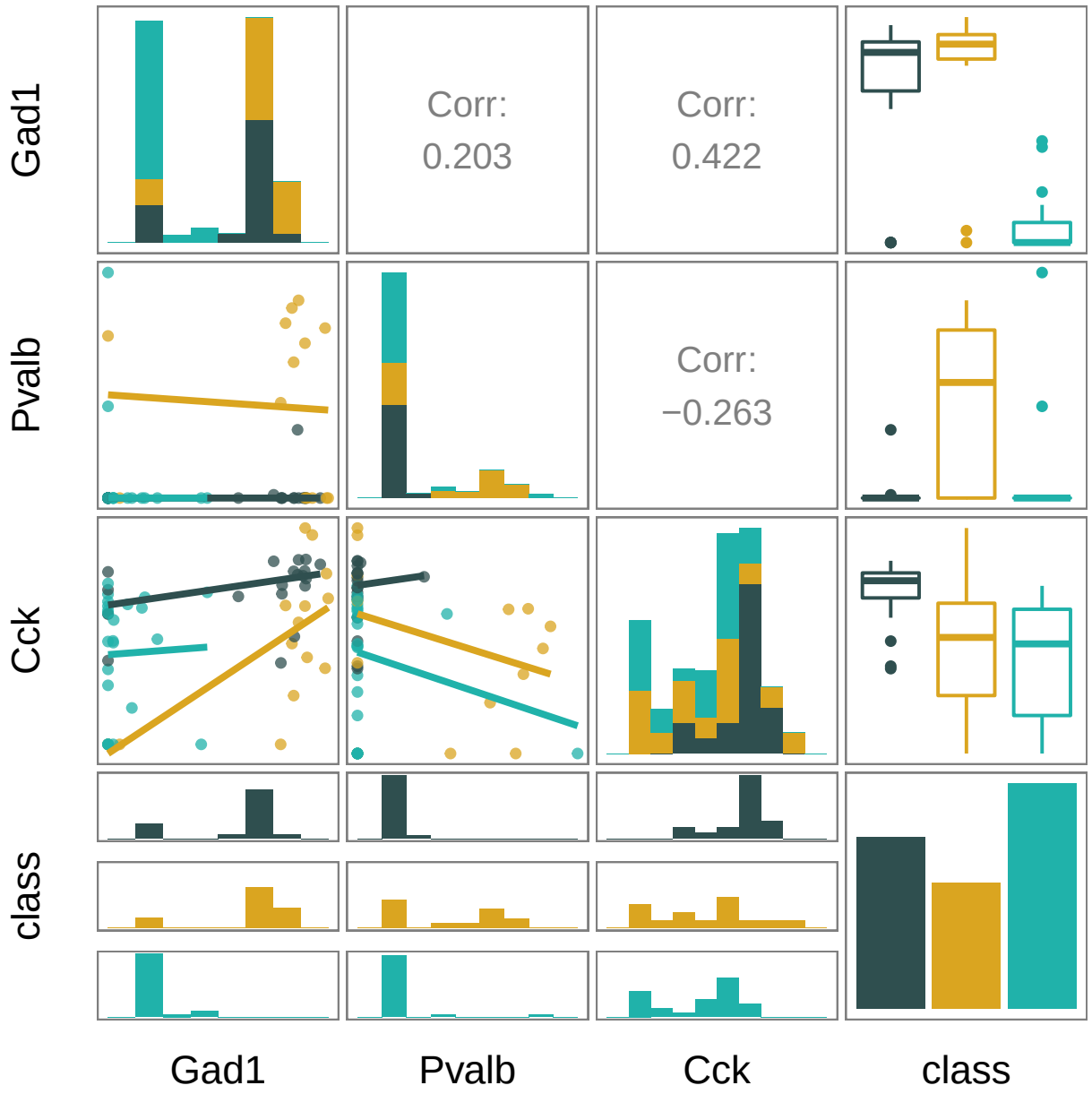

```
ggo = ggplot(anscombe, aes(x=x, y=y))
ggo = ggo + facet_wrap(~ set)
ggo = ggo + geom_point()
ggo = ggo + geom_text(aes(label=text), data=lmDescriptions(anscombe))
ggo = ggo + stat_smooth(method=rlm)
```

Local regression

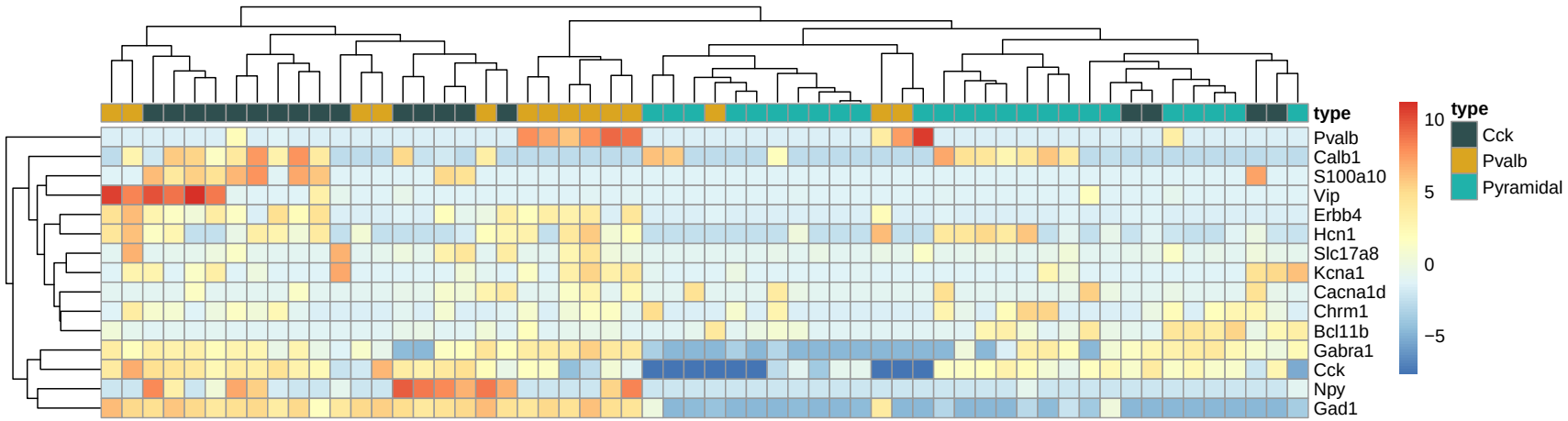



```
ggo = ggplot(anscombe, aes(x=x, y=y))
ggo = ggo + facet_wrap(~ set)
ggo = ggo + geom_point()

ggo = ggo + stat_smooth(method=loess, method.args=list(deg=1))
```

GSE75386 scatterplot matrix (a.k.a. pairs plot)

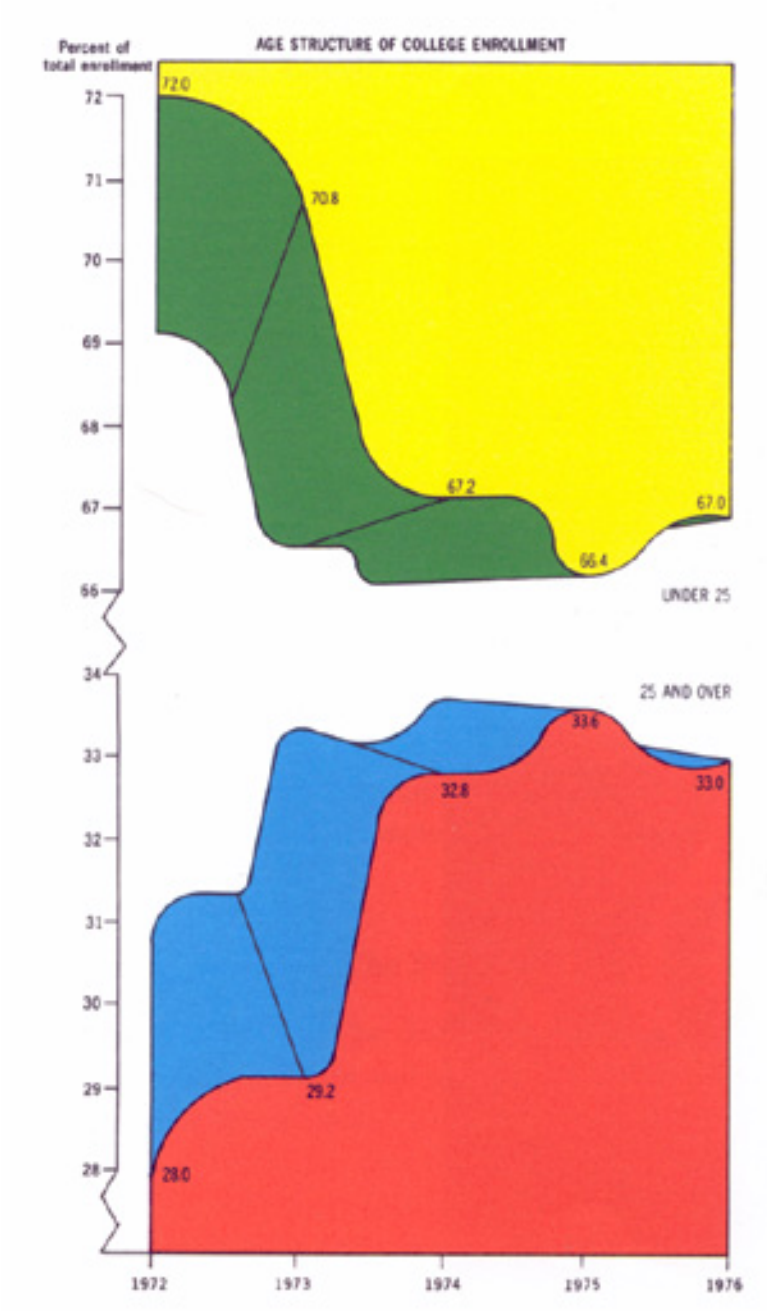


Clustered heatmap

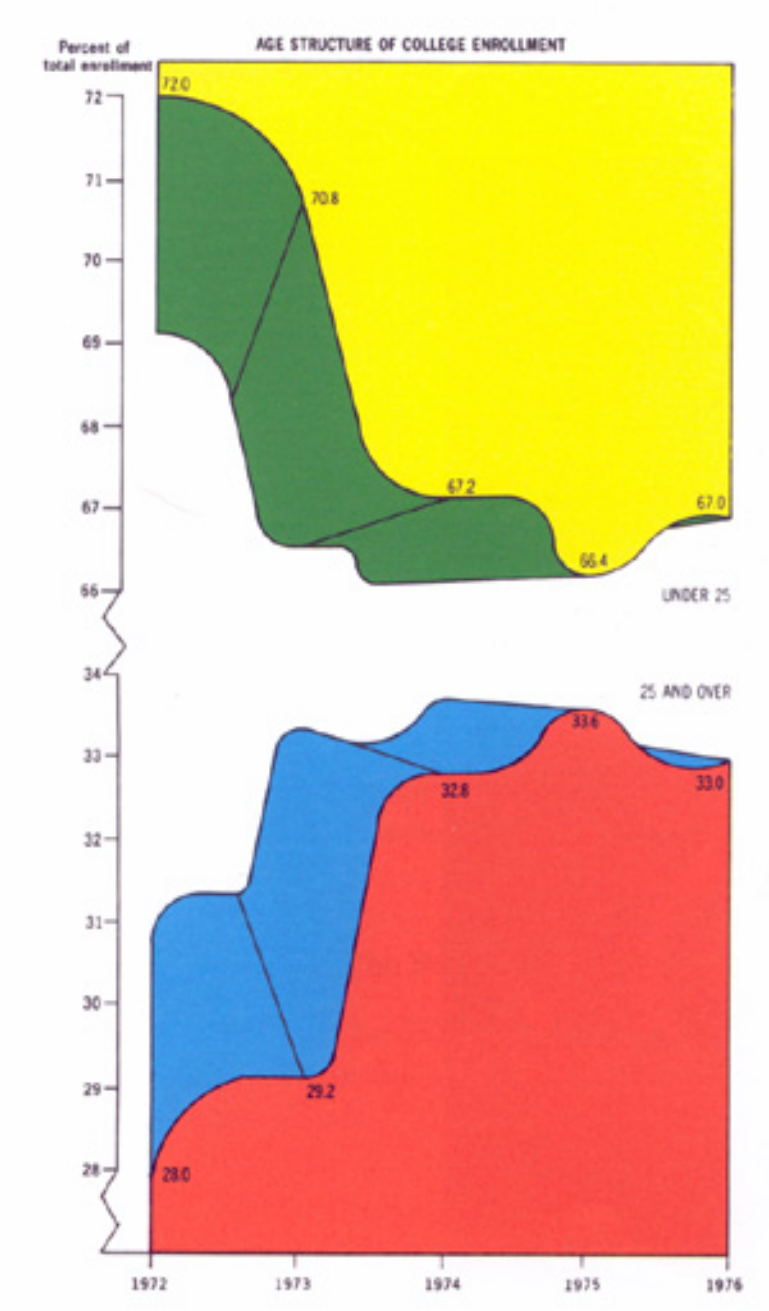



```
pheatmap(
    heatmapData,
    annotation_col = data.frame(
        row.names = colnames(heatmapData),
        type = simpleType[colnames(heatmapData)]
    ),
    annotation_colors = list(type=c(
        Cck = 'darkslategray',
        Pvalb = 'goldenrod',
        Pyramidal = 'lightseagreen'
    )),
    cluster_method = 'mcquitty',
    show_colnames = FALSE
)
```

Worst graph ever?



Worst graph ever?



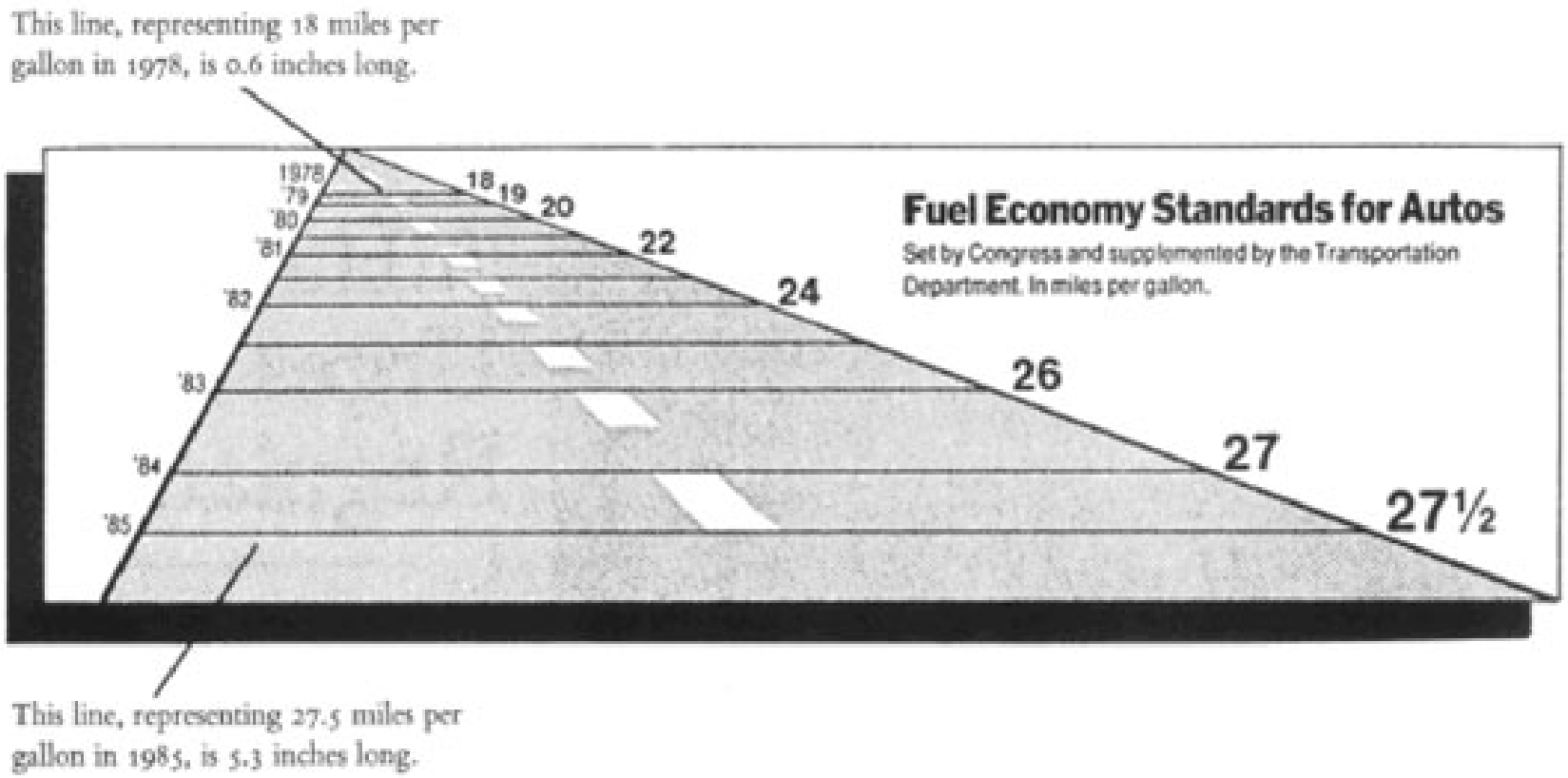
Tufte:

"Graphical excellence . . . gives the viewer

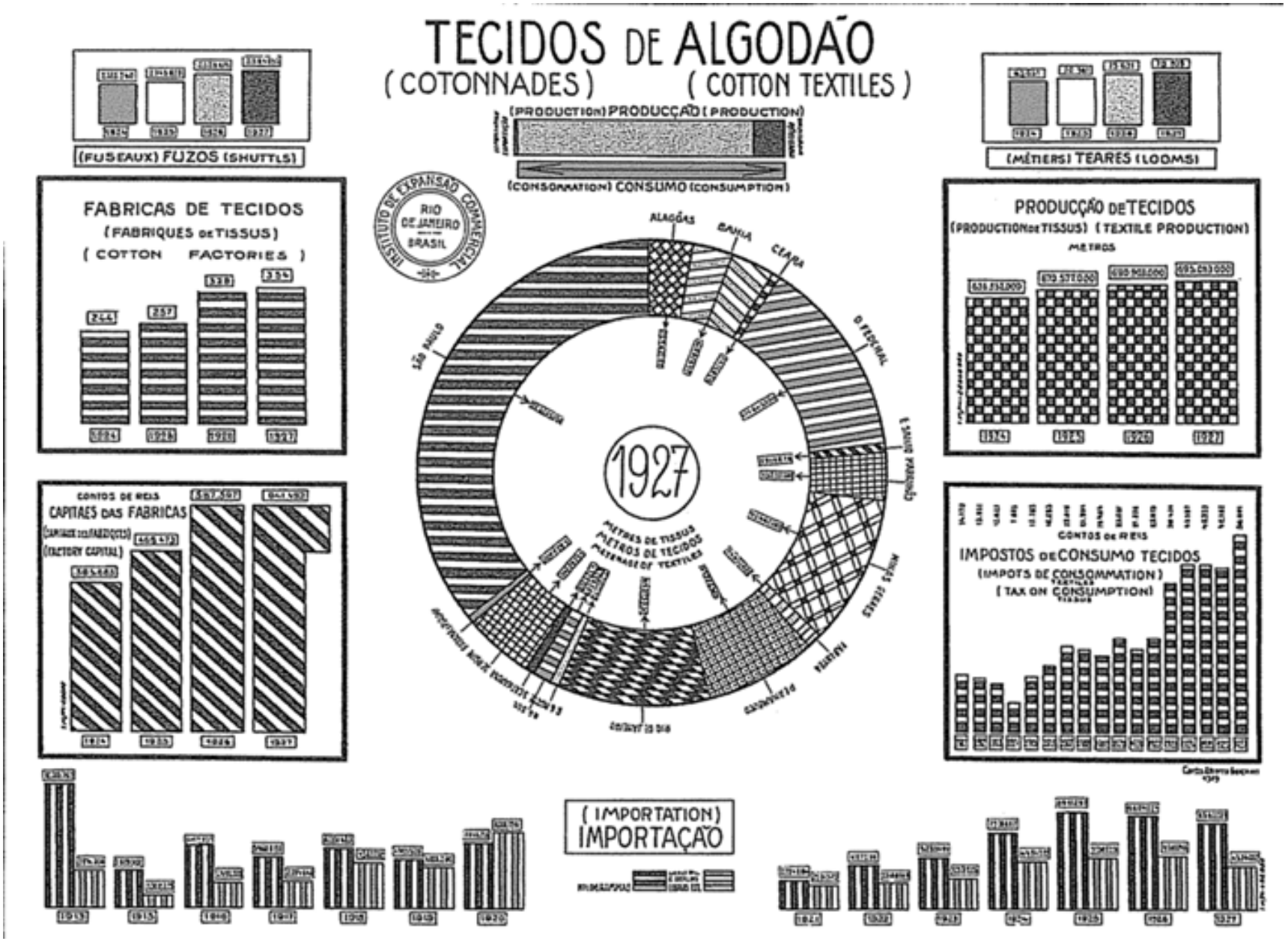
- the greatest number of ideas
- in the shortest time
- with the least ink
- in the smallest space."

When in doubt, remove ink/pixels which do not directly map data

This one's not so good either. . .



Moire effects



Moire effects