



























### Au menu...

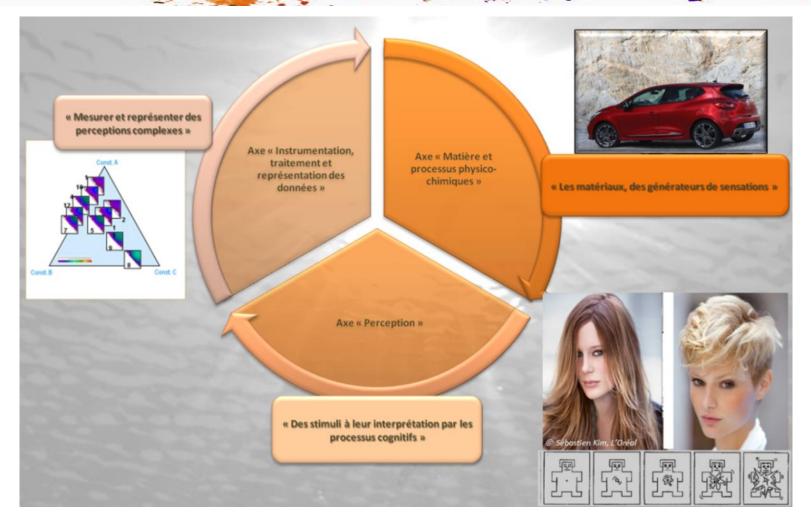
- D'où je parle
- La couleur, avec ou sans R
  - ...est un concept
  - rappel pour les débutants ?
  - du côté de la physique : R peut-il être « physico-réaliste » ?
- La couleur pour utilisateurs pressés : les catalogues
- La couleur pour bricoleurs : les grammaires
- Quelques applications spéciales : plongée dans la boite à outils R
  - émuler, capturer, échanger, customiser

Institut Mines-Télécom

Quelques références



### D'où je parle : ingénierie d'aspect





### D'où je parle : positionnement thématique

- Cartographie d'espaces "conceptuels"
  - analyse multivariée / multitableaux
  - "space embedding" / "mapping" / "alignement"
  - analyse morphologique
  - analyse de trajectoires
  - analyse sémantique
- Ingénierie des représentations visuelles
- Contributeur (modeste) à plotrix

Institut Mines-Télécom

Package de "GPS couleur" en préparation

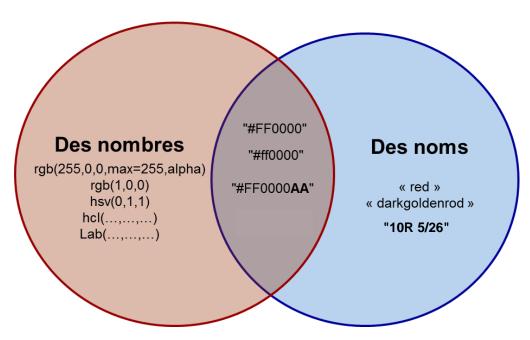
### La couleur, avec ou sans R, c'est...

- Le résultat d'un processus physique et psycho-physique
- Un sous-espace conceptuel (cognitif mais aussi industriel, normatif...)
- Un outil de communication visuelle, verbale et symbolique
  - Pour le marquage (identifier, grouper)

- Pour la quantification (échelles d'intensité)
- Pour la représentation fidèle (physico-réalisme)
- Pour l'amplification (animer, décorer)... à déconseiller ici (anti-Tufte!)
- Mais pour R?



### Mais la couleur, pour R ce n'est "que "



- supportés par des vecteurs ou des fonctions
- "workhorses": grDevices, colors() et colorRamp()
- détails: <a href="http://www.uv.es/conesa/CursoR/material/UsingColorInR.pdf">http://www.uv.es/conesa/CursoR/material/UsingColorInR.pdf</a>

6ème Rencontres R – Anglet, 28-30 juin 2017



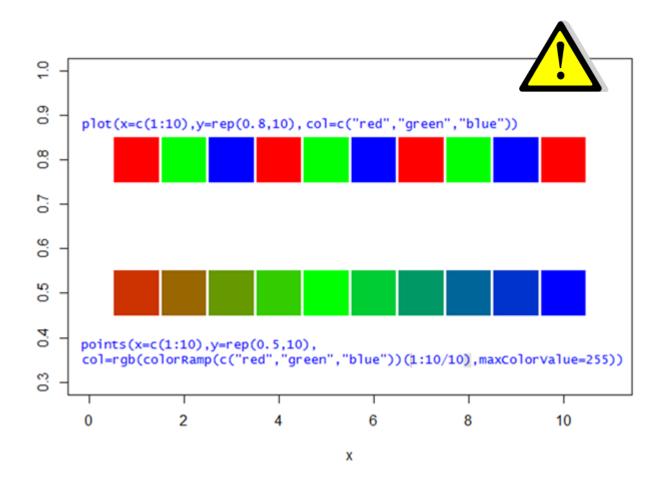
### Pour les débutants : vecteurs et rampes

#### **Vecteurs:**

à valeurs[i] correspond vecteurDeNomsCouleurs[i]

#### Rampes:

à valeurs[i] correspond FonctionCouleur(valeurs[i])

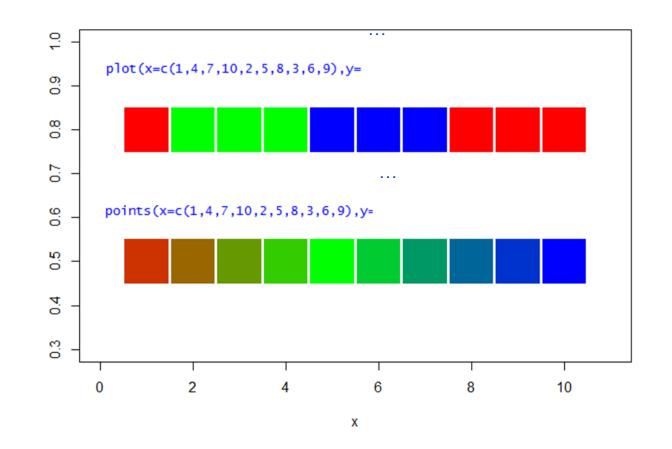




### Pour les débutants : vecteurs et rampes

**Vecteurs:** donc sensibles à l'ordre

Rampes: donc insensibles à l'ordre





8

## Pour les physiciens :

### R peut-il être physico-réaliste...



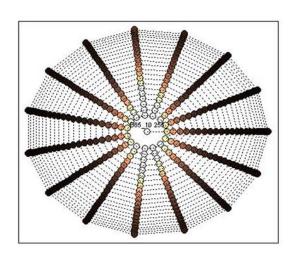


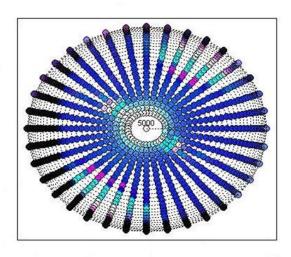
### Physico-réalisme : pour en faire quoi ?







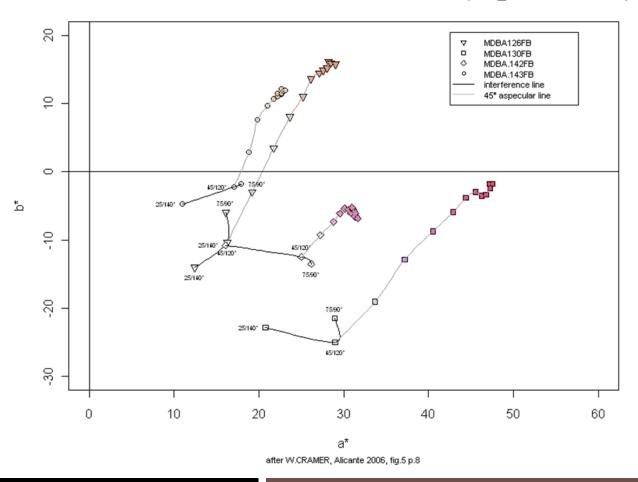






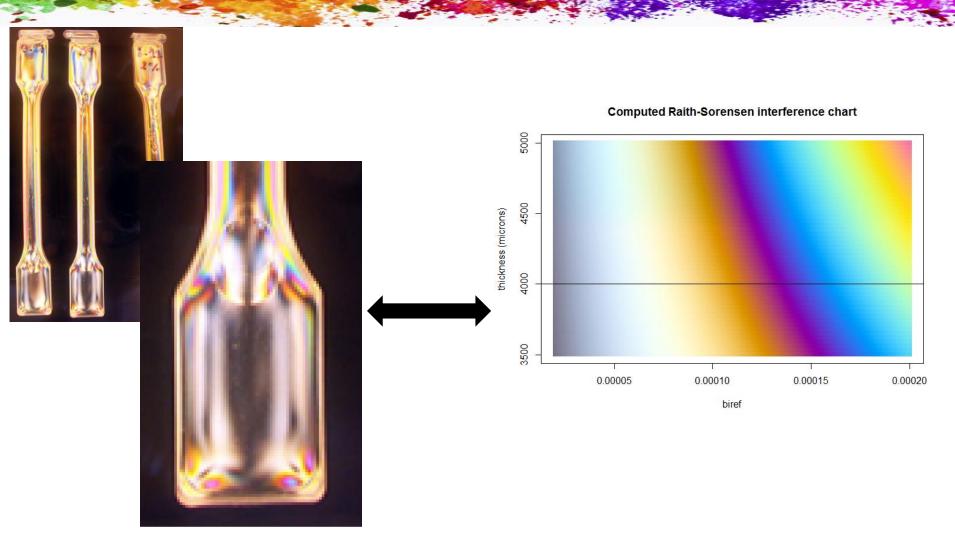
### Physico-réalisme : pour en faire quoi ?

#### MDBA126FB / MDBA130FB / MDBA.142FB / MDBA.143FB (D65\_10, ill. level = 500)



11

## Physico-réalisme : pour en faire quoi ?



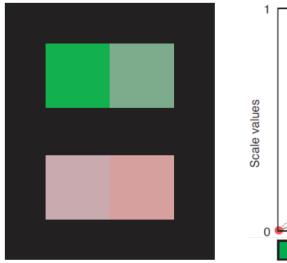


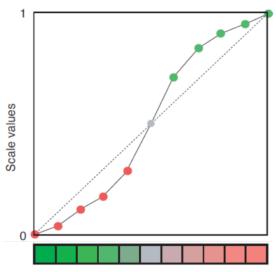
### Physico-réalisme : pour en faire quoi\* ?

\* voir aussi CRAN Task View: Psychometrics

Journal of Vision (2003) 3, 573-585 http://journalofvision.org/3/8/5/ Maximum likelihood difference scaling Department of Psychology and Center for Neural Science, Laurence T. Maloney New York University, New York, NY, USA **Joong Nam Yang** NASA Ames Research Center, Moffett Field, CA, USA  $\bowtie$ We present a stochastic model of suprathreshold perceptual differences based on difference measurement. We develop a maximum likelihood difference scaling (MLDS) method for estimating its parameters and evaluate the reliability and distributional robustness of the fitting method. We also describe a method for testing whether the difference measurement model is appropriate as a description of human judgment of perceptual differences in any specific experimental context. Keywords: sensory magnitude, proximity, similarity, difference scaling, salience

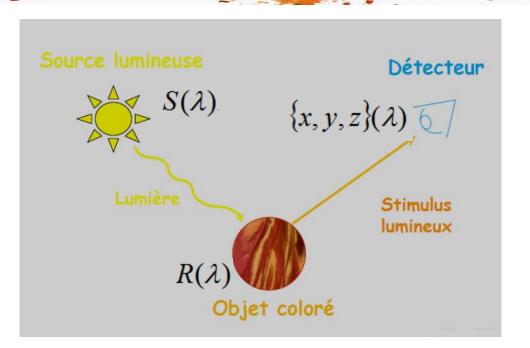
« Quelle paire est la plus différente ? » Épreuve dite « 2AFC » (choix forcé à 2 alternatives)







### Physico-réalisme à la sauce colorimétrique



$$\{X,Y,Z\} = k \int_{400}^{700} S(\lambda)R(\lambda)\{x,y,z\}(\lambda)$$
 $X Y Z$ 
objectif 29.38 24.54 14.25
réalisation 29.36 24.51 14.20

6ème Rencontres R – Anglet, 28-30 juin 2017

- R peut-il reproduire ce système ab initio?
- oui... et pour cela le « meilleur » package est colorSpec
- autres candidats : colorscience, pavo... mais attention





### La logique de colorSpec

1 – Décrire un couple source / récepteur (comportant éventuellement des filtres, etc...)

2 – spécifier un matériau qui servira d'étalon (ici, cas particulier d'un réflecteur parfait)

PRD <- neutralMaterial( 1, wavelength(D65.eye) )

3 - Calibrer le récepteur pour qu'il fournisse des coordonnées standard (penser au facteur k...)

6ème Rencontres R – Anglet, 28-30 juin 2017



### La logique de colorSpec (suite)

#### 4 – Calculer XYZ (avec product) puis transformer en RGB (avec RGBfromXYZ)

```
colorSpec.XYZ.RAL <- product( colorSpec.RAL, D65.eye, wave='auto' )</pre>
colorSpec.RGB.RAL<- RGBfromXYZ(colorSpec.XYZ.RAL, 'sRGB')</pre>
```

Institut Mines-Télécom

#### 5 – Adapter à la représentation écran (avec DisplayRGBfromLinearRGB par exemple)

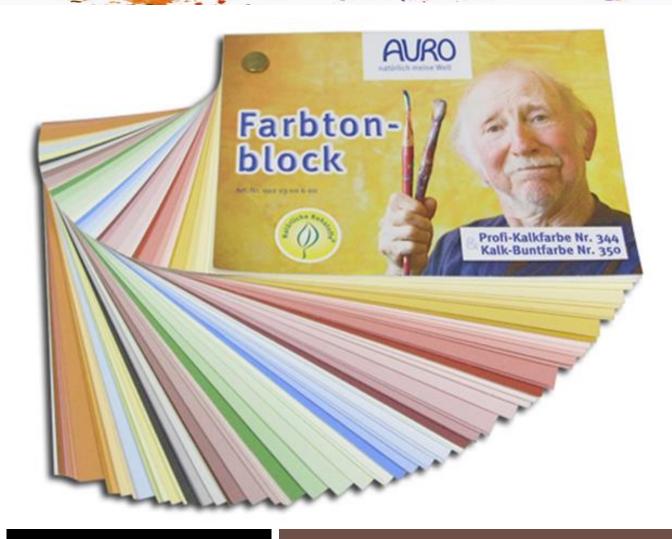
DisplayCoords<-DisplayRGBfromLinearRGB(colorSpec.RGB.RAL[c("RAL1000", "RAL1004", "RAL6000", "RAL6017"),], gamma='sRGB')







### Pour l'utilisateur pressé : couleurs sur catalogue

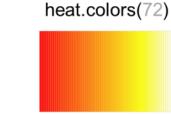




### L'offre de base : grDevices

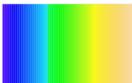
### palettes

rainbow(72)



colorRampPalette(

terrain.colors(72)



c("red", "orange", "blue"), space = "rgb")(72)





topo.colors(72)



cm.colors(72)



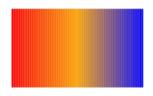
gray.colors(72)

(JND)

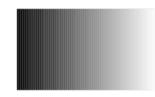
rampes [0,1]



colorRamp(c("red","orange","blue")) (0:72/72)



gray(0:72/72)





### RColorBrewer: "data driven"

- introduit l'approche "sémantique" : qualitative / séquentielle / divergente
- copyright (2002 C. Brewer, M. Harrower, The Pennsylvania State University)
- http://colorbrewer2.org
- palettes courtes, n<13

Institut Mines-Télécom

accès au catalogue au moyen d'une seule fonction : brewer.pal(n, name)



19

### RColorBrewer: "data driven"

#### qualitatives

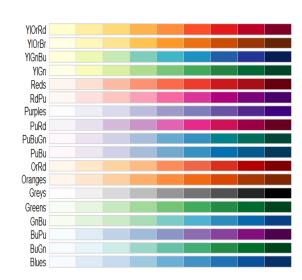


découpage discret

10/05/2017

- sans ordre intrinsèque
- catégorielles (facteurs)

#### séquentielles



- valeurs de "basses" à "hautes"
- valeurs de "inintéressantes" à "intéressantes"
- combiner faibles variations de teinte et fortes variations de clarté ou de saturation

6ème Rencontres R – Anglet, 28-30 juin 2017

#### divergentes

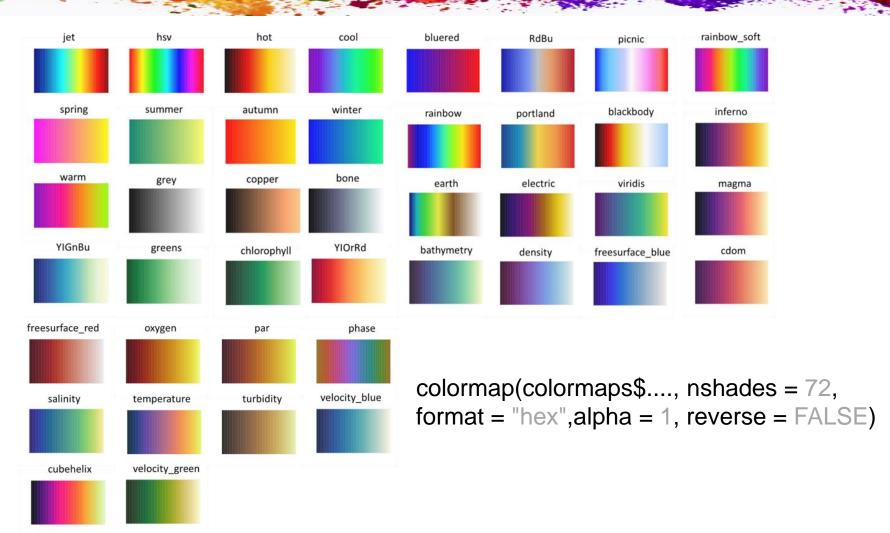


- Intérêt pour tout l'intervalle de variation
- point-milieu bien défini à valoriser
- Résidus, corrélations, probabilités



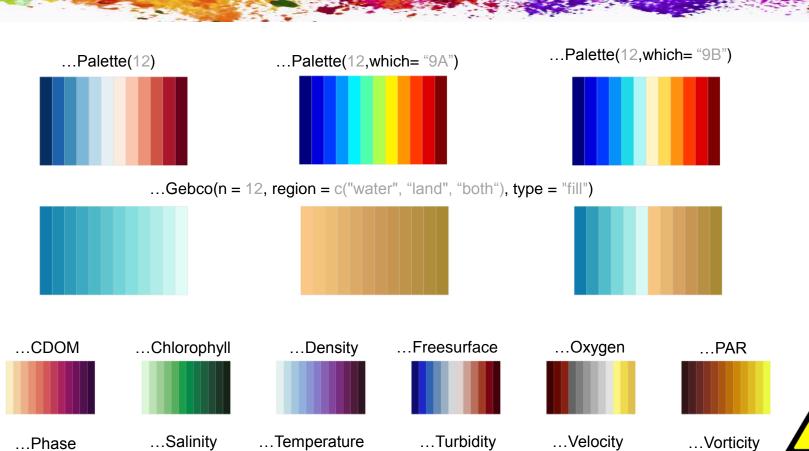
### colormap : la liste colormaps

6ème Rencontres R – Anglet, 28-30 juin 2017





### collection orientée océanographie : oce::oce.colors...



...Turbidity



...Velocity

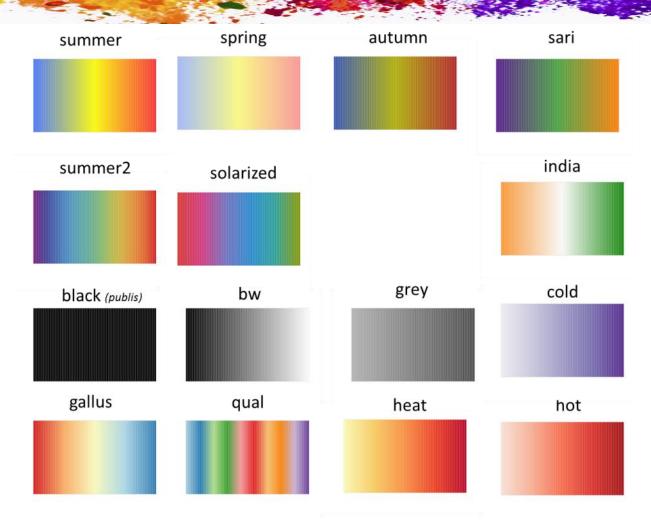
...Phase

...Salinity

Institut Mines-Télécom

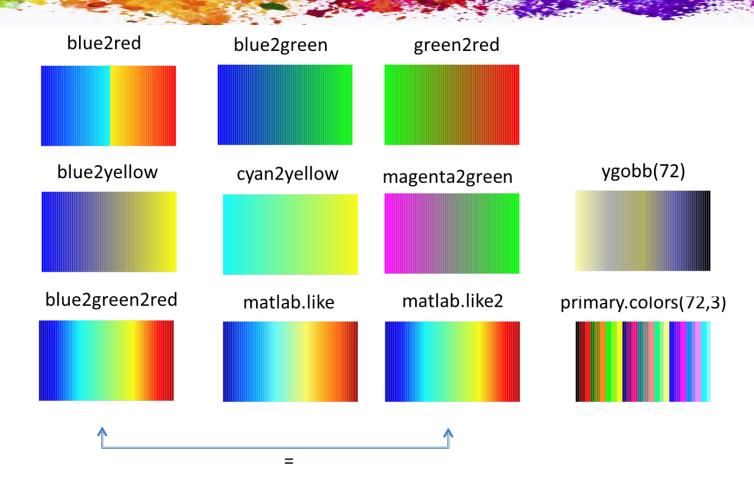
...Temperature

### Momocs::col



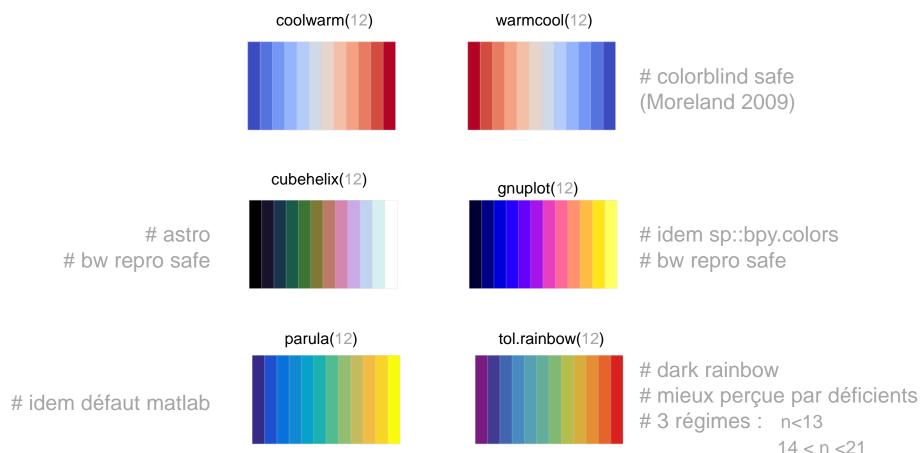


# colorRamps



plus une forme grammaticale spéciale (qu'on verra plus loin)







n > 21

25

### Maximiser les différences perçues : BacArena

COlpal1 : 269 couleurs "les plus distinctes de toutes les précédentes"





colpal2: 20 couleurs "optimalement distinctes"



colpal3: 22 couleurs "avec contraste maximum" (Kelly, Color Eng. 1965)



colpal4: 26 couleurs (Zeileis 2009, "Escaping RGBland")

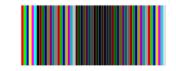


**COlpal5**: 64 couleurs selon un contributeur de stackoverflow (une coquille pour i=47 dans l'actuelle version la rend inopérante)



6ème Rencontres R – Anglet, 28-30 juin 2017

colpal6: 64 autres couleurs distinctes



### Maximiser les différences perçues : Polychrome

**Kelly.Colors**: 22 couleurs à comparer à BacArena::colpal3 (Kelly, Color Eng. 1965)

**glasbey.colors**: 32 couleurs "bien séparées" (Glasbey et al., Color Research and Application, 32, 304-9)

**green.armytage.colors**: 26 couleurs "pour un alphabet" (Green-Armytage, Colour: Design and Creativity, 2010; 10:1–23)

palette36.colors: 36 couleurs avec  $\Delta E_{L^*u^*v^*} \sim 40$  (Carter & Carter, Applied Optics, 1982; 21(16):2936–9)

**alphabet.colors**: 26 couleurs mêlant les approches de Carter et Green-Armytage

+ un ensemble de fonctions pour tester les palettes









rancurves(alphabet)
ranpoints(alphabet)
uvscatter(alphabet)
luminance(alphabet)
plothc(alphabet)
p3d(alphabet)



# Encore de fortes différences perçues : broman



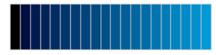


# Echelles perceptuelles linéaires : hexbin

LinGray (20, beg=1, end=92)



BTC (20, beg=1, end=92)



LinOCS (20, beg=1, end=92)



heat.ob (20, beg=1, end=92)



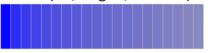
magent (20, beg=1, end=92)



plinrain (20, beg=1, end=92)



BTY (20, beg=1, end=92)



(chaque pas est une JND)



### Colorblind safe: la palette viridis

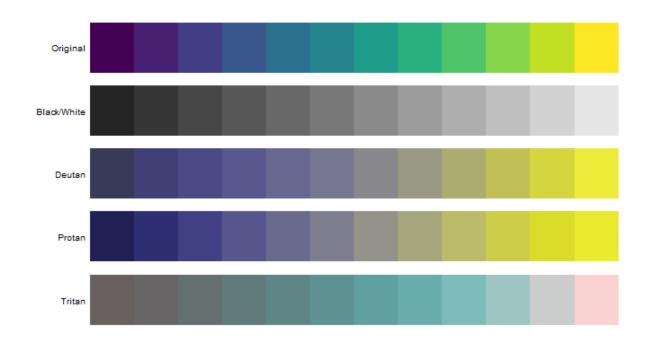
- n couleurs régulièrement espacées
- basée sur Matplotlib (python)
- dite "parfaitement uniforme" perceptuellement :
  - sous sa forme initiale
  - une fois convertie en niveaux de gris ("publication ready")
  - pour la plupart des types de DVC

- où la trouver?
  - pals::viridis(n,...)
  - colormap::colormap(colormaps\$viridis,nshades,...)
  - viridis::viridis\_pal(...)(n) et viridisLite::viridis(n,...)
  - parfois palette par défaut (heatmaply, dendextend, pdp, oce, IMIFA...)



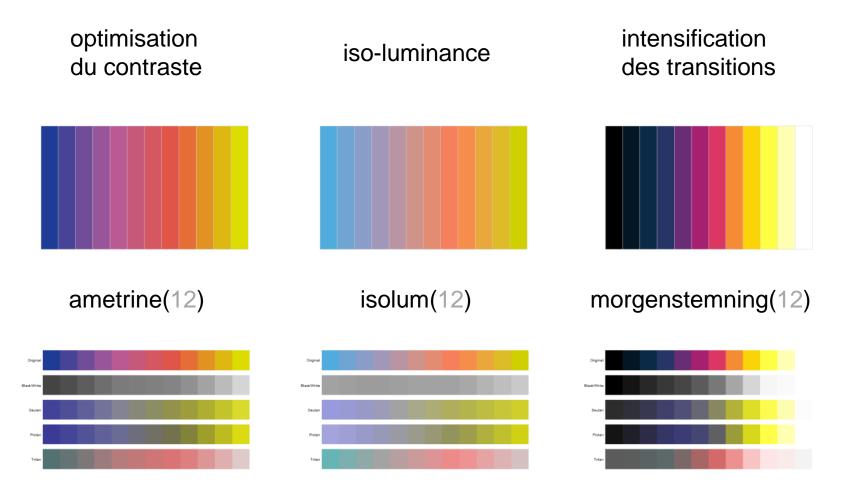
### Colorblind safe: la palette viridis

pals::pal.safe(pals::viridis(12))





### Colorblind safe: morgenstemning

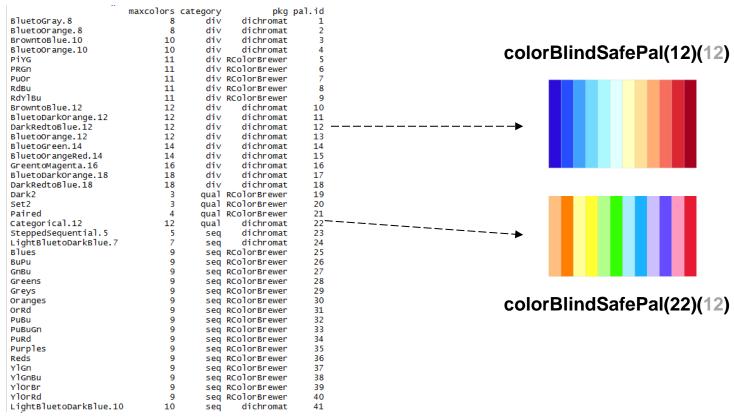




### Colorblind safe: biovizBase

#### "dichromat et RColorBrewer couplés"!

#### colorBlindSafePal()



Institut Mines-Télécom



33

### Quelques autres curiosités ...pas toujours fiables

■ colorr : palettes courtes (n ≤ 6) maillots et écussons de sport (anglo-saxons)

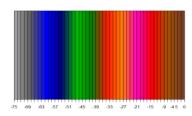


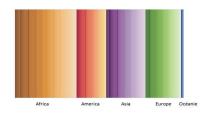


echogram : palette adaptée pour dB

**gapminder**: palette géopolitique (une teinte par continent, gradient pour les états)

Redmonder: palettes Microsoft (avec une sémantique imitée de ColorBrewer)



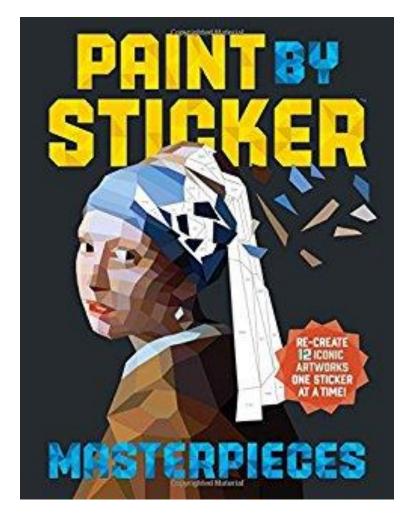






### Pour les bricoleurs.

les générateurs de palettes





### Les grammaires classiques

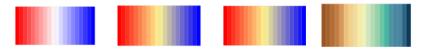
### colorspace

- espace HCL (hue / chroma / luminance)
- approche « Kansei » (dynamique / harmonique / froid / chaud)



### berryFunctions

- pas de fonction pour les palettes qualitatives
- dépasser la limitation du nombre dans ColorBrewer?
- ambiance par défaut « jaune / rouge / bleu / brun »

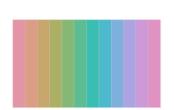




### colorspace

#### "qualitatives"

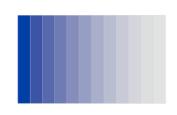
rainbow\_hcl(n, c = 50, l = 70, start = 0, end = 360\*(n-1)/n, gamma = NULL,...)



# h=seg(start,end,length.out=n) # voir [Zeileis 2009] pour usage

#### "séquentielles"

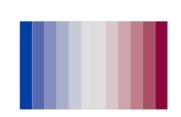
sequential\_hcl(n, h = 260, c. = c(80, 0), I = c(30, 90), power = 1.5, gamma = NULL,...)



# c. pour distinguer de c(...,...)

#### "divergentes"

diverge\_hcl(n, h = c(260, 0), c = 80, l = c(30, 90), power = 1.5, gamma = NULL,...)



# length(h) = 2# existe aussi en version hsv



### berryFunctions

#### "qualitatives"

pas implémentées, même si techniquement :

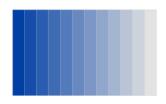
#### "séquentielles"

#### "divergentes"

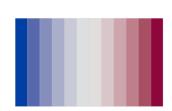








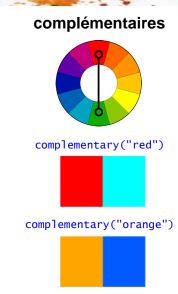
# colors : NULL ou vecteur comme dans colorRampPalette() ou brewer.pal()

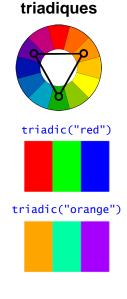


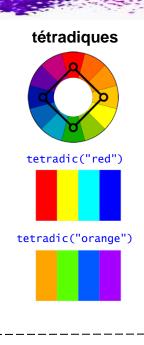


### Les alternatives - Géométriques / Circulaires

## analogues adjacent("red") adjacent("orange")

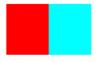


















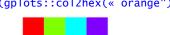


triadic(gplots::col2hex("orange")) tetradic(gplots::col2hex(« orange"))



tetradic(gplots::col2hex("red"))







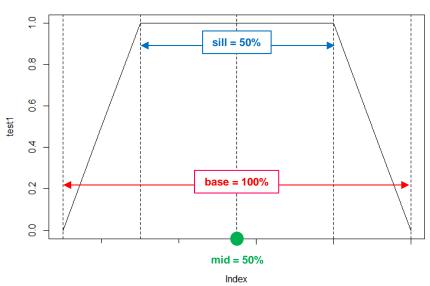


### Les alternatives - colorRamps

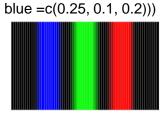
#### Objectif : créer une palette par "mélange" de 3 "primaires"

rgb.tables(n, red = c(mid, sill, base), green = c(m<sub>a</sub>, s<sub>a</sub>, b<sub>a</sub>), blue = c(m<sub>b</sub>, s<sub>b</sub>, b<sub>b</sub>)))

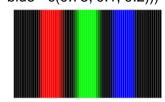








red = 
$$c(0.25, 0.1, 0.2)$$
  
green =  $c(0.50, 0.1, 0.2)$   
blue = $c(0.75, 0.1, 0.2)$ )



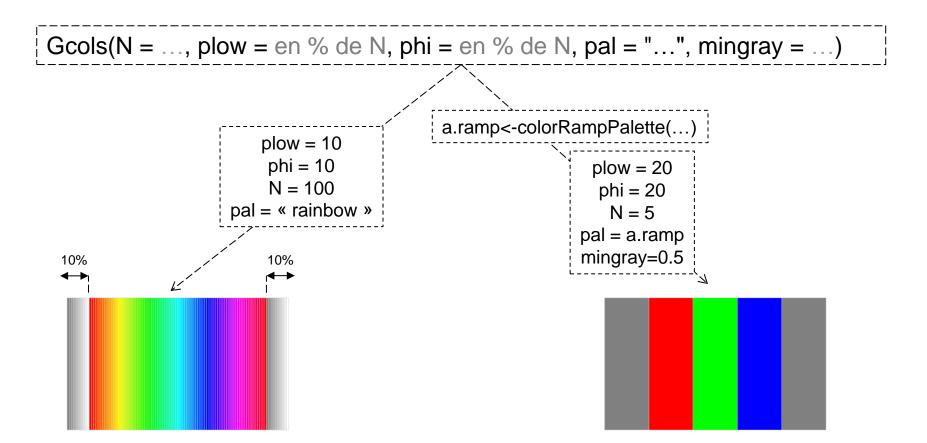
red = 
$$c(0.85, 0.33, 1)$$
,  
green =  $c(0.50, 0.5, 1)$ ,  
blue = $c(0.15, 0.33, 1)$ 





40

### Les alternatives - RPMG::Gcols

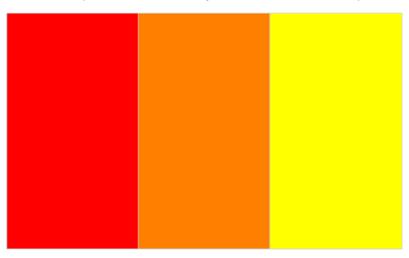




### Les alternatives - DescTools

#### Objectif: "mixer" deux couleurs en proportions variables

MixColor(col1="red", col2="yellow", amount1 = 0.5)



MixColor(col1="red", col2="yellow", amount1 = seq(1,0,by=-0.1))

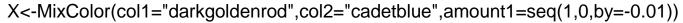


MixColor(col1="darkgoldenrod", col2="cadetblue", amount1 = seq(1,0,by=-0.1))



#### Les alternatives - DescTools

#### Intéressant à associer aux fonctions du package colorpatch





 $\# \operatorname{length}(X) = 101$ 

FindUniformSequence(colorspace::hex2RGB(X),n.out=15,delta=5,col.dist.fun=DistColorFun("LAB"))



FindUniformSequence(colorspace::hex2RGB(X),n.out=8,**delta=10**,col.dist.fun=DistColorFun("LAB"))





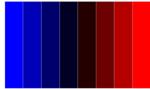
### Les alternatives - dimRed::mixColornRamps

#### Objectif : représenter 1 à 3 dimensions de variation par un mélange de rampes

#### **1D**

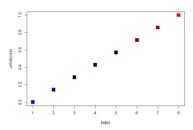
cols <- list(seq(0, 1, length.out = 8))
mixed <- dimRed::mixColor1Ramps(cols)</pre>

#### pal(mixed)



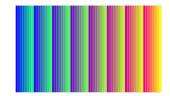
plot(unlist(cols), col = mixed, pch = 15, cex=2)

10/05/2017

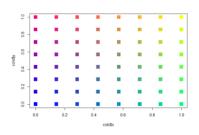


#### 2D

#### pal(mixed)



plot(cols\$x, cols\$y, col = mixed, pch = 15,cex=2)

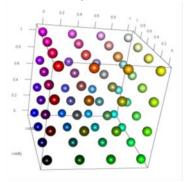


#### 3D

#### pal(mixed)



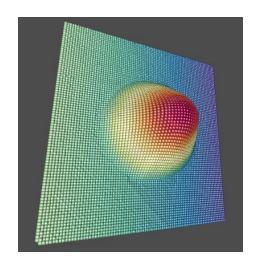
rgl::plot3d(cols\$x, cols\$y, cols\$z, col = mixed, type="s", radius=0.05)





### Passer de palettes quelconques aux rampes

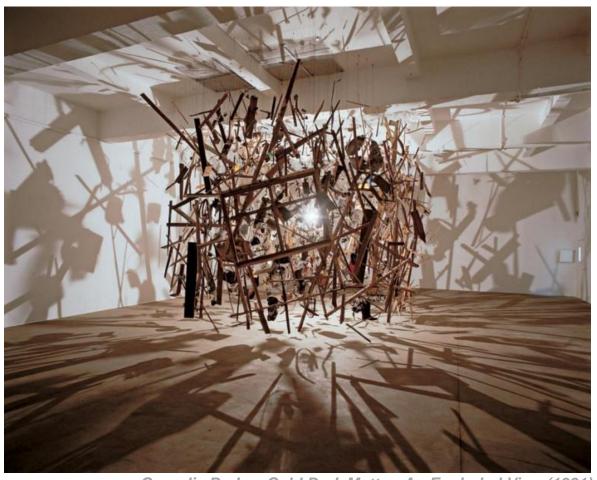
```
[1] "#9E0142" "#D53E4F<sup>"</sup> "#F46D43"
            Une palette
                                                   [,1] [,2] [,3] [,4]
  DescTools::HexToRgb(...)
                                           [1,] 0.6196078 0.003921569 0.2588235
                                           [2,] 0.8352941 0.243137255 0.3098039
             t(...) / 255
                                           [3,] 0.9568627 0.427450980 0.2627451
                                           [4,] 0.9921569 0.682352941 0.3803922
                                           Γ5.1 0.9960784 0.878431373 0.5450980
     colorspace::sRGB(...)
     class(…) ← "sRGB "
                                              function (value)
colorpatch::ColorRgbFun(...,
                                                 yy <- apply()
                                                 return(coerce.fun(colorspace::sRGB(R = yy$R$y,
xmin = ..., xmax = ...,
                                                 G = yySGSy, B = yySBSy)))
coerce.fun = colorspace::hex)
                                              <br/>
<br/>
de: 0x0000000539f4f90>
```





<environment: 0x000000004f86f810>

### Usages spéciaux enfouis dans le meccano R



Cornelia Parker, Cold Dark Matter: An Exploded View (1991)

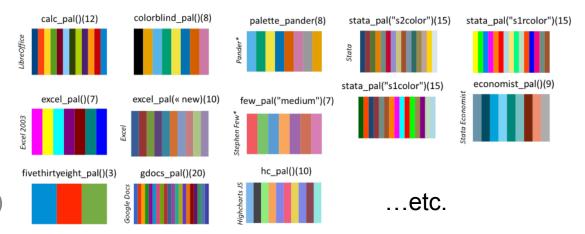


46

### Émulation de chartes - logiciels ou sites web

#### ggthemes

- LibreOffice
- Excel
- Stata
- Highcharts
- GoogleDocs
- Pander
- Tableau (pas montré)



- **GSIF**: palettes du SIG SAGA
- émulation matlab :
  - dans matlab : jet.colors() et multiline.plot.colors()
  - dans colorRamps : matlab.like() et matlab.like2()
  - dans squash : jet()
- oaColors : palettes tirées du site Open Analytics ( www.openanalytics.eu )



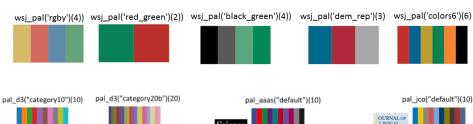
### Emulation de chartes - journaux et cartoons

#### ggthemes

Wall Street Journal

#### ggsci

- D3js.org
- Science
- Journal of Clinical Oncology
- The Lancet
- Nature Reviews
- **UCSC Genome Browser**
- GenePattern
- The Simpsons
- **Futurama**
- Rick and Morty











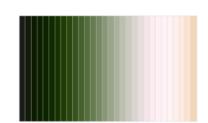








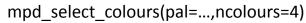
### Extraire des couleurs - rPlotter



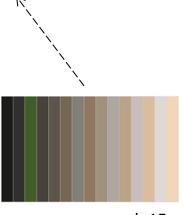


Maximiser la distance perceptuelle

interpolate\_colours(inp\_col=...,n\_col=30)



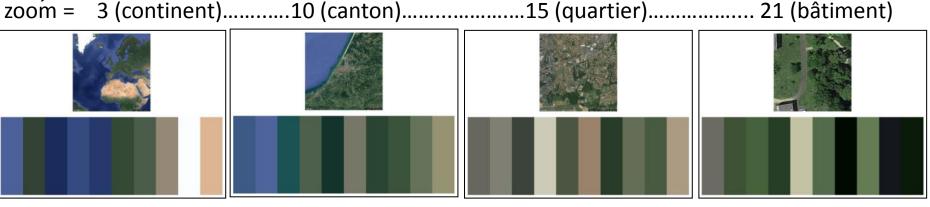




num\_col=15

extract\_colours("http://www.letudiant.fr/static/uploads/mediatheque/ETU\_ETU/7/1/182371-uppa-pau-patio-866x495.jpg",num\_col=...)

### Extraire des couleurs – earthtones



Résultat : list [[" pal "]] chr [number\_of\_color] [[" map "]] chr [1:1280, 1:1280]



### Extraire des couleurs – RImagePalette

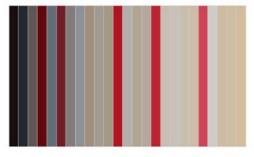
#### img



image\_palette(img, n=10)[order(...)]



image\_palette(img, n=25) )[order(...)]

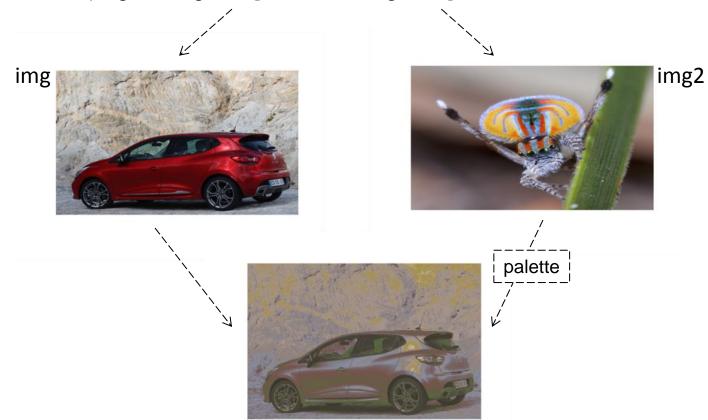


Utilisation de l'algorithme de « median cut »



### Et les échanger – RimagePalette

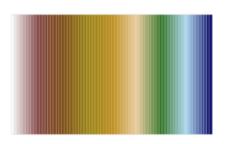
switch\_colors(target\_image=img, source\_image=img2,source\_colors=10,smoothness=100)



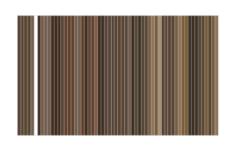


### Palettes orientées Sciences de la Terre

- NOAA (https://www.ngdc.noaa.gov/mgg/global/global.html)
  - marmap::etopo.colors()



aqp : quantitative pedology (Munsell)

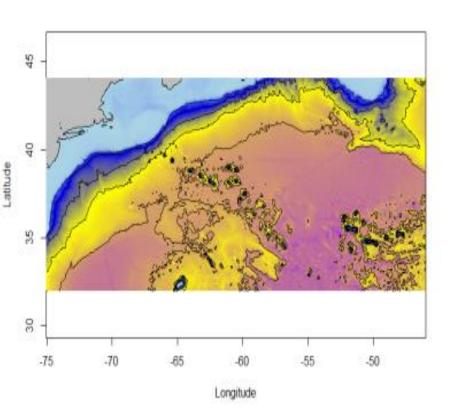




53

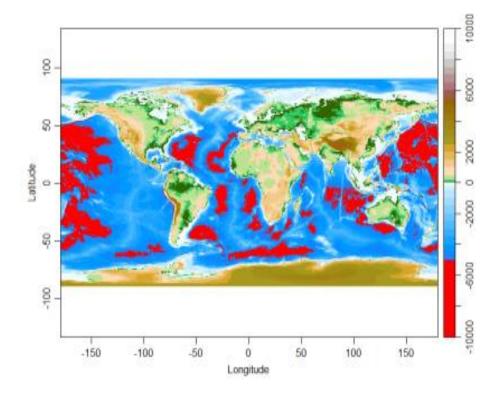
### Customisation maximale en cartographie

#### marmap::palette.bathy()



Institut Mines-Télécom

#### oce::colormap()





### Customisation maximale généraliste

#### squash::makecmap()

# x : vecteur des valeurs à "coloriser" # n : nombre de couleurs (approx.)

# breaks : les breaks ou une fonction les générant

# symm : booléen, symétrie autour de 0 ?

# base : pour une échelle log, la base; NA pour échelle linéaire

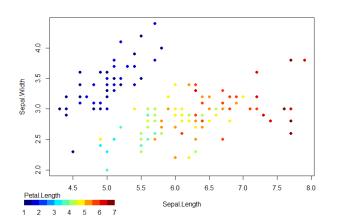
# colFn: la fonction générant la palette

# col.na: la couleur codant NA.

# right : booléen; intervalles fermés à droite ?

# include.lowest : booléen, ajouter une valeur égale à la limite

inf. (ou sup. si right = FALSE)?



Institut Mines-Télécom

#### squash::mtapply()

# x, y, z : vecteurs, ou une matrice # FUN: fonction à appliquer à z.

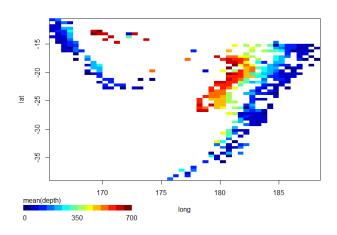
# nx, ny : nombre d'intervalles sur x et y (approx.).

# xlim, ylim: limites en x et y

# xbreaks, ybreaks : Breaks selon x et y # right : booléen; intervalles fermés à droite ?

# include.lowest : booléen, ajouter une valeur égale à la

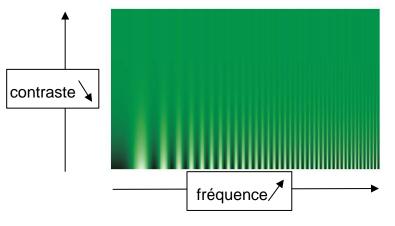
limite inf. (ou sup. si right = FALSE)?



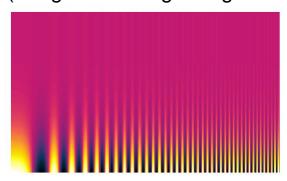


### Tester la visibilité des structures : pals

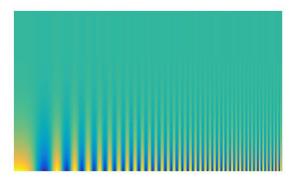
pal.csf(oce::oce.colorsChlorophyll)



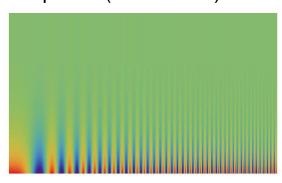
pal.csf(morgenstemning::morgenstemning)



pal.csf(parula)



pal.csf(tol.rainbow)





### Gérer les déficiences visuelles

- Utiliser des palettes « colorblind safe » (déjà vues)
  - morgenstemning
  - biovizBase
- Choisir pour optimiser la perception des différences
  - qualpalr::qualpal
- Tester ses palettes
  - LSD::daltonize # retourne palettes simulée et optimisée
  - rPlotter::simulate\_colours # ne retourne rien
  - pals::pal.safe # ne retourne rien



### qualpair

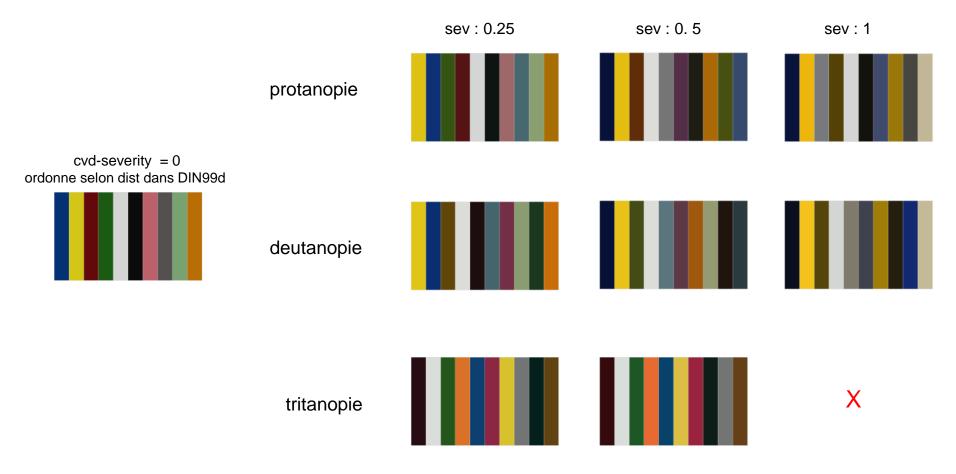
#### Objectif: choisir dans une collection une palette "la mieux adaptée possible" à une anomalie de vision

- $list(h{=}c(h_{min}{,}h_{max}),\ s{=}c(s_{min}{,}s_{max}),\ l{=}c(l_{min}{,}l_{max}))$  $\in$ [-360,360] ∈[0,1]  $\in [0,1]$
- character vector ∈ {"pretty", "prettydark", "rainbow", "pastels" }
- matrice de couleurs, dans l'espace sRGB avec D65
- data.frame (coercible en matrice)

- red-blindness: protanopia ("protan", sev=1)
- red-weakness: protanomaly ("protan", 0<sev<1)
- green-blindness: deuteranopia ("deutan", sev=1)
- green-weakness: deuteranomaly ("deutan", 0<sev<1)
  - blue-blindness: tritanopia ("tritan", sev=1)
  - blue-weakness: tritanomaly ("tritan", 0<sev<1)
    - normal : sev = 0



# qualpalr

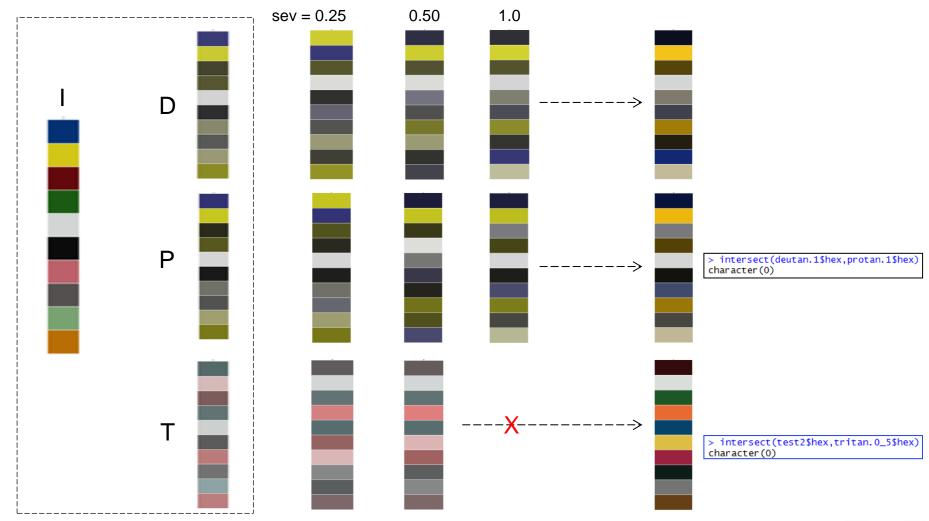




Institut Mines-Télécom

### Tests de palette avec rPlotter

6ème Rencontres R – Anglet, 28-30 juin 2017

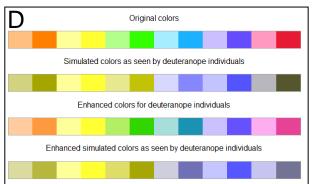




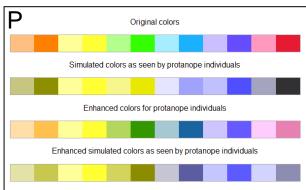
### Test de palette avec LSD::daltonize()

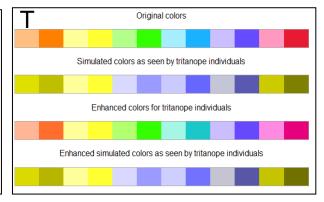


colorBlindSafePal(22)(12)



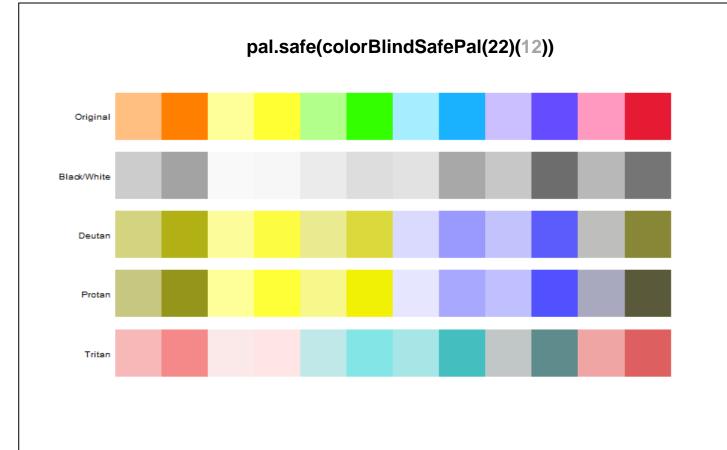
Institut Mines-Télécom





### Test de palette avec pals







Institut Mines-Télécom

62

### Quelques références parmi tant et tant...

Zeileis A., Hornik K., Murrell P. (2009) "Escaping RGBland: Selecting Colors for Statistical Graphics". Computational Statistics & Data Analysis, 53, pp. 3259- 3270. doi:10.1016/j.csda.2008.11.033

**Griggs** B. (2014) "The end of the rainbow? An exploration of color in scientific visualization". *Bachelor Thesis*, University of Oregon.

**Tol** P. (2012) "Colour Schemes". SRON technical note 09-002 version 2.2, 16 p.

**Silva** S., Sousa Santos B., Madeira J. (2011) "Using color in visualization: a survey". *Computers & Graphics*, **35**, pp. 320-333

**Nelson** J. (2015) "Truthful Mapping". https://www.linkedin.com/pulse/truthful-mapping-john-nelson (version révisée de l'original de 2011)

Jenny B., Vaughn Kelso N. (2007) "Color Design for the color vision impaired". Cartographic perspectives, 58, pp. 61-67. http://colororacle.org/resources/2007 JennyKelso ColorDesign hires.pdf

Wijffelaars M. (2008) "Synthesis of color palettes". Master's Thesis, Technische Universiteit Eindhoven, 91 p.

**Brewer** C. (1994) "Color use guidelines for mapping and visualization". Chap.7 in "Visualization in modern cartography", A.M. MacEachren & D.R. Fraser Taylor eds. Pergamon

Bernhard J. (2012) "Principles of Data Visualization". http://stat.pugetsound.edu/courses/class13/dataVisualization.pdf

Institut Mines-Télécom



### Quelques sites d'exploration de palettes...

6ème Rencontres R – Anglet, 28-30 juin 2017

- ColorSchemeDesigner (<a href="http://colorschemedesigner.com/csd-3.5/">http://colorschemedesigner.com/csd-3.5/</a>)
  - grammaire « géométrique »
- Colour Lovers (http://www.colourlovers.com/)
  - orienté palettes qualitatives
  - catalogue palettes et couleurs
- Kuler (https://color.adobe.com/fr/create/color-wheel/)
  - grammaire « géométrique »
- ColorBrewer (<a href="http://colorbrewer2.org">http://colorbrewer2.org</a>)
  - grammaire classique







