Plotten mit Perl 5
Streudiagr.

scatter plot
Gitterdiagramm.

mesh plot
Polar-Koor. polar plot
Kreisdiagramm

Pie chart
Säulendiagramm

bar chart
Points Scored Per NBA Game

- Carmelo Anthony
- Dwyane Wade
- Deron Williams
- Brook Lopez
- Damian Lillard

Box plot
Plots with Perl

1. Math: data prep.
2. Design: Colors & Shapes
3. Code: Tech. & Modul PP, PL, gnu., gg., plotly
To PDL or not
more talks:

Plots with Raku
Chart::Data
Part 1: Math

Data
Data
Data Analysis
Low Lvl. Prep.

PLACE DIRTY TOWELS HERE
Create Data Set

1. document all meta data sources, constrains author, time
   (keep a raw copy/link)
Create Data Set

2. converting/unifying format
(but label source)
(CVS, Serial, dat, PDL:

Fast Raw, HDF
Create Data Set

read human: CSV, TSV, .dat YAML
compact: fastraw, HDF, Serial, SQL
structured: Serial, JSON, YAML
2. unifying format (CVS, Serial, dat, PDL) readable label (code lvl), consist., descriptive names
Create Data Set

3. Codd’s 3rd normal form

I. table = observation type
II. row = observation / ID
III. columns = var/dep. val
Create Data Set

3. melt (tidy data)

Hadley Wickham paper

new 3 column table:

ID, 'property', value
Create Data Set

3. reshape data
at least consistent

   can be used as is
4. cleaning:
(but keep a copy)
null, bad value, consistent ranges
4. measure data MB of sets, dimensions, types (add to meta data)
5. descriptive statistics: extreme (min max), location (avg. median), distribut. (var.), correlation
Create Data Set

5. descriptive statistics:
Statistics::Basic  App::St
PDL::Stats::Basic
5. descriptive statistics:

save data /
save generating code
Data Prep.

What data sets we have?

What info we show?

For Whom?
Choices

**filter**: faulty, range, outlier, representative

**split**: by property/range
summarize: avg, median, dev, modus, skew, corr feature?
Select Dimension

1. x = 2, y = 4, z = 7
2. x = 4, y = 5, z = 7
3. x = 1, y = 8, z = 7
4. x = 9, y = 3, z = 7
Variance

\[ \text{var}(X) = 2.5 \]
\[ \text{var}(Y) = 1.5 \]
\[ \text{var}(Z) = 0 \]
use Statistics::Basic qw/ :all /;

my @z = $data->{,z'}

say variance( @z );
Select Dimension

Principle Component Analysis
use Statistics::PCA;
my $pca = Statistics::PCA->new;

$pca->load_data( { format => 'table',
  data => [ [..], [..], ..], } );

$pca->pca();

$pca->results('eigenvalue');
# list of array (eigenvector)
found most interesting data

What if not enough?
Or variance too high
smooth plot

Kernel Density Estimation.
use Statistics::KernelEstimation;
my $s = Statistics::KernelEstimation->new_gauss();
$s->add_data( $_->{'x'}, $_->{'y'} ) for @data;
   # data   weight    width
my $w = $s->default_bandwidth();
for my $x ( min .. max ) {
    my $y = $s->pdf( $x, $w );
}
Kernel Dens. Est.

smooth curve in foreground

range of actual data as background shadow
KDE of variance
On every x position:

new $\_->\{,'y'\} = y - \text{pdf}(x)$

$\text{pdf}(x) = \text{range height}$
use Math::Spline;

my $spline = Math::Spline->new(@x, @y);
my $y = $spline->evaluate($x);
Beziers vs KDE
polar coord.
Log. Skalierung.

log scaling
Log. Skalier.

log scaling
artificial log

no log($<=1$)

1. label: x, value: log(x)
2. value: log(x+1)
3. mirror x$<0$
artificial log

log enlarges lower part (nearer to 0)

enlarge upper with square
big libs have log scaling
but e.g. gnuplot only $x > 0$
Part 2: Design

Color, Shape, Organisation
Color Theory
What web ppl use
canva.com/colors/color-palettes/
w3schools.com/colors/coolors.co
Basic Rules

frame: scales / ticks / label

normal data

highlighted data points (label)
(high saturation)

background grid, meta data label
range marker
Color Space of Brain

**Hue** which color on rainbow
red .. violett  = 0..1

**Saturation** color .. grey

**Lightness** white .. color .. black
Color Space of Brain

Hue  which color on rainbow
red .. violett  = 0..1

Saturation  color .. grey

Value  color .. black
Transparency

manage plot overlap with alpha ~ 0.8
darker colors for compensation
don‘t use mixed colors
Transparency

gnuplot : rgb "#ααRRGGBB"

PLplot : 0000FF_0.1

don‘t use mix colors
HSL Space
gnuplot

hsv2rgb(h, s, v)

(not in Chart::Gnuplot)

$chart->command ("gnuplotCommand");
HSL → RGB

```perl
use Convert::Color;

my $red = Convert::Color->new("hsl:0,1,0.5");
my ($r,$g,$b) = $red->rgb;

# values: 0..1, 0..1, 0..1
```
use Convert::Color;

my $red = Convert::Color->new("hsl:0,1,0.5");
my ($r, $g, $b) = $red->rgb;

# $red->convert_to('rgb')->{rgb};

# values: 0..1, 0..1, 0..1
use Convert::Color;

my $red = Convert::Color->new("hsl:0,1,0.5");

my ($r,$g,$b) = $red->rgb;

$red->convert_to('rgb8')->hex;

# values: FF00000
use Convert::Color;

my $red = Convert::Color->new("hsl:0,1,0.5");
my ($r,$g,$b) = $red->rgb;
$red->dst_hsl($color);

# distance is HSL space
HSL Space

H, S, L = 3 dim. of information

1. use one color for all data
2. vary 1 dim if it contains info
   H - quality, S, L - quantity
3. vary more if good reasons
HSL Space
Gradients
Gradients don’t have to be linear to highlight interesting threshold
Color vs 3D
Color vs 3D

Surface Plot of Test3

Test3
- 90 - 96
- 84 - 90
- 78 - 84
- 72 - 78
- 67 - 72
- 61 - 67
- 55 - 61
- 49 - 55
- 43 - 49

Stress = 0.2287
Color + 3D
## Color + Digits

### Confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>airplane</th>
<th>automobile</th>
<th>bird</th>
<th>cat</th>
<th>deer</th>
<th>dog</th>
<th>frog</th>
<th>horse</th>
<th>ship</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplane</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>automobile</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>bird</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>cat</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>deer</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>frog</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>horse</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>ship</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>truck</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
1,5 scales

- India: 29%
- Africa: 18%
- Asia/Oceania*: 17%
- Americas: 13%
- China: 11%
- Africa/Middle East: 9%
- Europe: 3%
- Others: 1%

*Excluding China

- India: 29%
- China: 13%
- Africa: 18%
- Asia/Oceania: 17%
- Americas: 11%
- Africa/Middle East: 9%
- Europe: 3%
- Others: 1%

*Excluding China

Country
- India
- Africa
- Asia/Oceania
- China
- Americas
- Africa/Middle East
- Europe
- Others
Pie vs Bar

Market Share

- Competitor A: 29.9%
- Competitor B: 19.0%
- Us: 18.7%
- Competitor D: 14.3%
- Competitor E: 12.7%
- All Others: 5.4%

Percentage of the Entire Market
Background fill colors that encode qualitative ranges like bad, satisfactory and good

Revenue 2017 YTD
(U.S. $ in thousands)

Quantitative Scale
Symbol marker that encodes the comparative measure

Bar that encodes the performance measure
composition

2008
- Johnson: 43
- Livingstone: 37
- Paddick: 10
- Others: 10

2012
- Johnson: 44
- Livingstone: 40
- Paddick: 4
- Others: 10
slightly better
good paper

Stephen Few

Practical Rules for using Colors in charts
Organisation

Color, Shape, Size, Pattern, Distance, Enclosure, Connectedness, Foregr.
What important quality you want to highlight?
Organisation

Steven’s Psychophysical Power Law: $S = I^N$

- Perceived Sensation vs. Physical Intensity
- Lines represent different stimuli: Electric Shock (3.5), Saturation (1.7), Length (1), Area (0.7), Depth (0.67), Brightness (0.5)
Data Types

quantity
ordinal
quality
Data Types

**quantity:** pos, length, area, thick, bright., saturat.

**ordinal:** axis order, font size, fore / background

**quality:** color hue, shape, (line)pattern, font
just enough data

which values to print out?
(or hide in tool tips)
Organisation

communicate relevant data without overburden
just enough data

all data points

cluster / groups

set summary
violine + box

The image shows a violin plot with box plots for different conditions:

- **Control**: The distribution is relatively narrow, with a median around 30.
- **Exp. A**: The distribution is slightly wider than Control, with a median around 32.
- **Exp. B**: The distribution is the widest, with a median around 34.

The X-axis represents the condition (Control, Exp. A, Exp. B), and the Y-axis represents the score, ranging from 15 to 50.
multi histog.
Strengths

violine area = mass

mult. hist.: detailed

margin: particulars
just enough data

too many plots vs.

one crammed
<table>
<thead>
<tr>
<th>Channel</th>
<th>Sessions</th>
<th>Bounce Rate</th>
<th>Pages / Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>24,127</td>
<td>46%</td>
<td>1.8</td>
</tr>
<tr>
<td>Organic Search</td>
<td>16,524</td>
<td>50%</td>
<td>1.9</td>
</tr>
<tr>
<td>Paid Search</td>
<td>13,429</td>
<td>53%</td>
<td>1.8</td>
</tr>
<tr>
<td>Referral</td>
<td>11,182</td>
<td>36%</td>
<td>2.5</td>
</tr>
<tr>
<td>Display</td>
<td>9,488</td>
<td>97%</td>
<td>1.0</td>
</tr>
<tr>
<td>Social</td>
<td>455</td>
<td>21%</td>
<td>2.4</td>
</tr>
<tr>
<td>Email</td>
<td>423</td>
<td>35%</td>
<td>2.0</td>
</tr>
</tbody>
</table>
4D plot

Sepal.Length
-0.12  0.87  0.82

Sepal.Width
-0.43 -0.37

Petal.Length
0.96

Petal.Width
Part 3: Code

PURE PERL
PL. GNU. PLUTIL
GGPL.PLOTLY SDL
Modules

PURE PERL

Plperl gnupl
ggpl.plotly SDL
PP Modules
Chart

uses GD for JPG / PNG

professional 90‘ies look

solid Docs in LaTeX PDF HTML
Chart

Composite Chart

Direction Demo

Dataset 1
Dataset 2
Dataset 3
nice OO Design:

Bars, Composite, Direction, ErrorBars, HorizontalBars, Lines, LinesPoints, Mountain, Pareto, Pie, Points, Split, StackedBars, (polar coor)
Chart
from ‘90 - last release 2015
Bundesamt für Kartographie und Geodäsie,
Geodätisches Observatorium Wettzell
CPANID: CHARTGRP
Finding Chart on cpanmetad failed (& website defunct)
Imager::Plot

imUgly.ttf (Verdana)
calles not existing sub

*::Axis::i_color_color_new
SVG::Plot
SVG::::Graph
SVG::Graph

no label

no file save

(pipe svg to stdout)
### SVG::Graph

<table>
<thead>
<tr>
<th>Glyph Name</th>
<th>Dimensionality supported</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1d</td>
</tr>
<tr>
<td>----------------</td>
<td>----</td>
</tr>
<tr>
<td>Axis</td>
<td>x</td>
</tr>
<tr>
<td>Bar Graph</td>
<td>x</td>
</tr>
<tr>
<td>Bubble Plot</td>
<td>x</td>
</tr>
<tr>
<td>Heatmap Graph</td>
<td>x</td>
</tr>
<tr>
<td>Line Graph</td>
<td>x</td>
</tr>
<tr>
<td>Pie Graph</td>
<td>x</td>
</tr>
<tr>
<td>Scatter Plot</td>
<td>x</td>
</tr>
<tr>
<td>Spline Graph</td>
<td>x</td>
</tr>
<tr>
<td>Tree</td>
<td>x</td>
</tr>
</tbody>
</table>
SVG::TT::Graph::Pie

„presentation quality“

removed first file line

title missing
Chart::Plot

My Graph Title

X label

Y label
Chart::Plot

of 2001, uses ano GD 2D only, few line styles autosize, 4 colors, (.anno)
GD::Graph

Nov 2016, uses GD

many chart types
autosize, many colors
but:
version 1.2
in 1999
GD::Graph::type->new(w,h);
$graph->set(%options);
$graph->plot([..],[..]..);
check /samples/save.pl
Example 42

A Lines and Points Graph, reading a CSV file
Example 42

A Lines and Points Graph, reading a CSV file

Y Label

1st 2nd 3rd 4th 5th 6th 7th 8th 9th

X Label

data set 1  data set 2
GD::Graph

image size ++
more margins
same data marker (7)
changed colors to own
GD::Graph

GD::Graph::colour::add_colour

('midblue', [70,90,210]);

$graph->set({
    markers => [7, 7],
    marker_size => 3,
    dclrs => [qw(midred ..)],
});
$graph->set( 
    [ markers => [7, 7],
      marker_size => 3,
      t_margin => 20,
      b_margin => 20,
      l_margin => 10,
      r_margin => 20, ]
);
3D Barcharts & Histograms
PDL::Graphs::PLplot

best main & doced
feature rich
easiest to install
PLplot

started 1986 by Sze Tan in Fortran-77
C version by Tony Richardson & al. in 90′
PLplot

Linux, Mac, Win
CGM, GIF, JPEG, PBM, PDF, PNG, PostScript, SVG, Xfig
GDI, GTK+, Qt, Tk, Wx, X
PLplot

low level command set

base fore other API
PDL is a slang

```perl
my $x = sequence(10);
my $y = $x ** 2;
pl->xyplot($x, $y);
```
PDL is a slang

```perl
my $y = pdl(2, 3, 4, 5);
my $y = pdl( @data );
$p1->xyplot($x, $y);
```
use PDL;
use PDL::Graphics::PLplot;
my $pl = PDL::Graphics::PLplot->new(
  DEV => "png", FILE => "..png",
  PAGESIZE => [700,600]);
$pl->xyplot($x, $y);
my $pl = PDL::PLplot->new(
    SUBPAGES => [1,2], # XY dir
    COLOR => 'RED',
);

$pl->histogram( $x, 10,
    BOX => [0, 10, 0, 3],
    SUBPAGE => 2);

$x = \text{pdl}(2,3,4,5);$
3D
```perl
use PDL;
use PDL::Graphics::PLplot;
my $pl = PDL::Graphics::PLplot->new( ... );

$pl->pl_cmd( "..." );
```
PDL::Graphics::PGPLOT

mostly interactive API
does not run
Graphics::PLplot

same API as PLplot
does not run
Graphics::Gnuplot

V0.23  2015 (bsd dev?)

error messages due
image conversion
Graphics::Gnuplot

1. **bei0(x)**
   - A plot of the function $\text{bei0}(x)$.
   - The graph shows oscillations with peaks and troughs.

2. **Hat function (3D)**
   - A 3D plot of the hat function.
   - The function peaks at the origin and diminishes away from it.
   - The color scale indicates the values of the function, ranging from 0 to 1.
PL < Gnu
January 2022!
founded 2007
out of an R package
modern optics & funct.
s*ton of dependencies (JS)

no PDL binding
no PDL binding
use PDL::Math
PDL as data store
Chart::Plotly::

Chart::Plotly::

Trace::***

all plot types
Chart::Plotly

Chart::Plotly::

Trace::***

very good JS doc
Chart::Plotly

march 2022!
get the file out

```perl
use Chart::Plotly::Image 'save_image';

my $plot = Chart::Plotly::Plot->new();
$plot->add_trace($bessel);

save_image( plot => $plot,
    engine => 'kaleido',
    file => 'plot.png',
    width => 500, height => 400, );
```
Util Modules

Color::Library
Convert::Color
Statistics::Basic
Statistics::PCA
Statistics::KernelEstimation
Math::Spline
Math::Bezier
slides: lichtkind.de

https://juiceanalytics.com/chartchooser/
https://canva.com/colors/color-palettes/
https://w3schools.com/colors/
https://coolors.co/
https://colorbrewer2.org/
https://github.com/d3/d3-scale-chromatic/
http://pdl.perl.org/
That's all

Thank You