Data Visualization

Data Science Summer School, July 15th 2021

Richard Traunmüller, University of Mannheim

I Brought Some Data…

X1	Y1	X2	Y2	X3	Y3	X4	Y4
57.61323	83.90517	52.87202	97.3432236	51.20389	83.339775	55.3846	97.1795
51.27439	82.81799	59.01414	93.5748749	58.97447	85.499817	51.5385	96.0256
50.75390	76.75413	56.37511	96.3051453	51.87207	85.829735	46.1538	94.4872
37.02118	81.95447	37.83920	94.3594437	48.17993	85.045120	42.8205	91.4103
42.88176	80.18477	39.88537	90.6346588	41.68320	84.017944	40.7692	88.3333
37.15571	84.95411	44.07740	84.1258545	37.89042	82.567490	38.7179	84.8718
38.73186	83.78936	31.49702	67.9125595	39.54897	80.812599	35.6410	79.8718
31.00052	82.57948	25.95260	73.0331802	39.64957	82.664536	33.0769	77.5641
25.98550	74.34939	36.43472	62.2940445	34.75060	80.011093	28.9744	74.4872
23.71457	75.09805	24.99748	75.4415588	27.56083	72.847824	26.1538	71.4103
23.07732	71.72054	32.24628	70.1105881	24.63554	71.610718	23.0769	66.4103
24.93914	72.58028	18.10947	53.7081108	20.95947	66.041496	22.3077	61.7949
17.89350	71.10969	21.79610	49.4995308	20.68915	62.721306	22.3077	57.1795
23.67308	64.87084	24.15049	47.1885338	19.28820	62.063061	23.3333	52.9487
23.74907	63.55717	28.60320	44.2237244	20.02450	61.342625	25.8974	51.0256
32.21518	65.60096	20.93200	51.6462364	35.46952	43.115887	29.4872	51.0256
29.43684	67.55453	35.43553	34.9610367	36.89433	47.706554	32.8205	51.0256

40.05009	71.88903	44.37738	57.6847382	39.05555	55.546974	35.3846	51.4103
37.57479	69.98875	46.31369	55.6855926	46.95708	65.240410	40.2564	51.4103
49.31047	67.16876	46.38607	51.0021629	37.31045	45.258751	44.1026	52.9487
47.34802	72.59161	42.49544	56.9036827	40.00967	60.986584	46.6667	54.1026
50.15741	71.16173	44.30051	58.1329575	48.01439	65.712822	50.0000	55.2564
58.21001	68.04454	46.98829	55.4206352	53.70378	66.389481	53.0769	55.6410
56.60891	70.40894	52.42152	47.8670044	63.06750	64.035004	56.6667	56.0256
52.83382	72.55189	64.09100	68.6509857	62.04803	63.845863	59.2308	57.9487
61.81858	76.11282	63.01687	68.2252579	59.83997	64.476761	61.2821	62.1795
61.72763	77.06924	67.30257	64.4134598	55.16094	65.237305	61.5385	66.4103
62.19225	76.15682	67.22596	62.4764099	61.27979	65.766403	61.7949	69.1026
58.11877	73.11897	50.30999	47.2678833	60.83492	64.603767	57.4359	55.2564
58.27041	35.59689	45.41264	53.8707962	61.52059	64.791855	54.8718	49.8718
59.76653	32.62125	54.02061	47.2698975	36.91654	41.095249	52.5641	46.0256
52.53553	27.17412	49.84202	50.5765533	38.50220	41.567154	48.2051	38.3333
50.89252	29.14528	50.71330	48.7722855	48.66437	30.680666	49.4872	42.1795
49.96755	28.46020	55.04556	45.9862709	50.28525	30.337923	51.0256	44.1026
51.34789	27.36426	37.54370	32.8840256	42.27633	34.527637	45.3846	36.4103
37.55511	24.53766	39.79590	28.4432182	54.03178	29.672348	42.8205	32.5641
39.19959	23.68050	28.52339	39.7966652	37.32936	39.602043	38.7179	31.4103
36.38272	25.23800	31.39965	39.2652626	41.38952	37.296055	35.1282	30.2564

41.10069	26.55261	39.24157	33.3614159	40.07467	34.623684	32.5641	32.1795
28.21781	28.28029	27.71406	42.2500534	35.34968	47.141071	30.0000	36.7949
33.79707	28.69975	32.96070	35.8231125	34.76370	47.624802	33.5897	41.4103
43.91939	27.79784	35.40474	31.9391613	37.02663	44.462292	36.6667	45.6410
44.41640	28.94769	51.64223	50.8383408	36.45557	40.791843	38.2051	49.1026
28.03760	26.80382	34.20762	33.5938072	35.53766	48.729385	29.7436	36.0256
31.68958	26.66640	36.52349	35.6710472	20.40895	32.203030	29.7436	32.1795
27.27961	19.25728	28.41920	21.3157482	23.49571	25.322468	30.0000	29.1026
31.22188	26.44086	26.16050	23.0422306	29.55754	21.364777	32.0513	26.7949
33.54581	20.08071	40.43533	27.6400719	33.00823	15.985071	35.8974	25.2564
39.73633	23.49481	44.54496	27.3622417	53.98040	29.350986	41.0256	25.2564
42.84184	20.33390	40.12373	30.0731621	52.23431	29.711674	44.1026	25.6410
47.09373	26.99347	42.18481	28.5736752	59.50308	30.669674	47.1795	28.7180
51.45366	26.02294	40.00266	27.4875107	41.16378	34.315758	49.4872	31.4103
49.36771	22.00558	57.12523	42.7390175	48.99304	32.030357	51.5385	34.8718
48.25348	29.27085	60.37574	41.3564568	59.26928	29.640701	53.5897	37.5641
61.00296	30.01531	55.10590	46.1490517	45.46918	33.831192	55.1282	40.6410
60.80656	27.86831	61.73727	38.5848427	62.69127	30.290375	56.6667	42.1795
59.06743	26.78828	55.74125	46.6535759	73.42867	48.577854	59.2308	44.4872
61.41463	31.03938	58.29641	41.5831985	70.84643	52.282253	62.3077	46.0256
68.93437	31.61989	57.04255	42.4187584	71.53902	45.521805	64.8718	46.7949

63.06524	34.23575	76.77561	55.2546005	67.62086	38.006557	67.9487	47.9487
72.27252	67.64445	71.88708	59.2669945	72.47095	51.122135	70.5128	53.7180
75.35988	75.42162	73.78838	58.6208115	64.81224	62.810917	71.5385	60.6410
72.83363	72.11469	75.13969	54.2445526	60.85368	65.499146	71.5385	64.4872
70.81042	79.35046	64.59632	65.9525146	67.78950	61.363701	69.4872	69.4872
45.80193	81.65826	46.90954	85.6022644	41.60956	83.848686	46.9231	79.8718
45.83502	82.74259	46.90510	81.5656433	53.00303	84.674797	48.2051	84.1026
40.07881	80.24718	49.80699	82.0850220	54.71417	84.043129	50.0000	85.2564
51.98293	82.20438	48.86324	83.1583557	44.29167	82.909439	53.0769	85.2564
57.33791	84.99952	66.74819	84.0048370	49.19172	85.876228	55.3846	86.0256
62.33623	83.24657	62.35977	90.2436600	53.10138	84.547661	56.6667	86.0256
58.50591	81.74558	47.37490	83.4715576	51.59985	84.819824	56.1538	82.9487
56.24794	83.51395	47.44647	82.6221237	54.37972	84.240356	53.8462	80.6410
55.40652	80.26667	53.38528	78.8711395	46.48077	83.518211	51.2821	78.7180
53.28397	84.26360	49.45271	79.9306030	53.17466	84.260567	50.0000	78.7180
44.67676	80.23829	54.80268	75.3639831	45.27200	85.237076	47.9487	77.5641
30.01126	68.14676	37.08825	60.8572273	36.03340	53.371689	29.7436	59.8718
29.74525	74.40453	32.37596	68.5108185	28.27119	72.840233	29.7436	62.1795
37.30995	72.58616	37.47701	63.7886848	25.05481	71.548599	31.2821	62.5641
61.13595	82.53757	52.40665	96.4254150	64.75889	82.315224	57.9487	99.4872
61.72062	85.70570	50.29704	99.6441803	63.14453	85.236694	61.7949	99.1026

63.60938	83.36003	61.56332	89.8641357	50.42468	85.174751	64.8718	97.5641
70.72372	87.15221	61.08656	89.2243576	70.64500	82.430916	68.4615	94.1026
75.00639	82.42583	59.23649	91.4811020	63.14905	83.946854	70.7692	91.0256
75.09255	83.43739	66.15401	84.1119385	62.82402	84.966187	72.0513	86.4103
70.32793	82.02936	67.23134	85.0556412	70.23687	82.171150	73.8462	83.3333
70.86333	79.40817	73.70991	78.2706146	70.04273	80.385025	75.1282	79.1026
75.77991	79.88620	76.58045	72.8915329	72.57063	80.971222	76.6667	75.2564
78.09366	77.75956	78.87658	71.5625305	75.13071	79.984093	77.6923	71.4103
76.74576	71.97336	77.44901	72.4473114	83.29391	70.778435	79.7436	66.7949
78.46461	69.74930	86.85371	65.5433273	79.66426	73.932312	81.7949	60.2564
76.74568	70.04406	79.80653	52.7231789	88.43211	64.624245	83.3333	55.2564
85.16168	65.51241	76.98827	54.2027855	89.11556	64.001503	85.1282	51.4103
91.80835	34.96560	79.47935	49.1410141	89.09220	57.768192	86.4103	47.5641
91.88745	34.76925	82.30416	46.9909020	91.72601	52.623352	87.9487	46.0256
92.24841	32.37716	90.70069	41.4667015	91.73554	48.970211	89.4872	42.5641
91.78253	34.38430	95.44349	36.1897011	91.50789	53.312653	93.3333	39.8718
96.08052	28.05360	94.99749	35.0953026	88.23900	31.477434	95.3846	36.7949
92.24790	28.07075	91.51785	36.8902626	88.53052	30.476030	98.2051	33.7180
57.66228	24.06061	62.29264	40.0295334	55.36516	30.445850	56.6667	40.6410
59.84692	26.17771	59.20348	41.2706795	62.56026	30.447136	59.2308	38.3333
64.06709	27.77650	65.25639	32.8541031	58.00667	30.253721	60.7692	33.7180

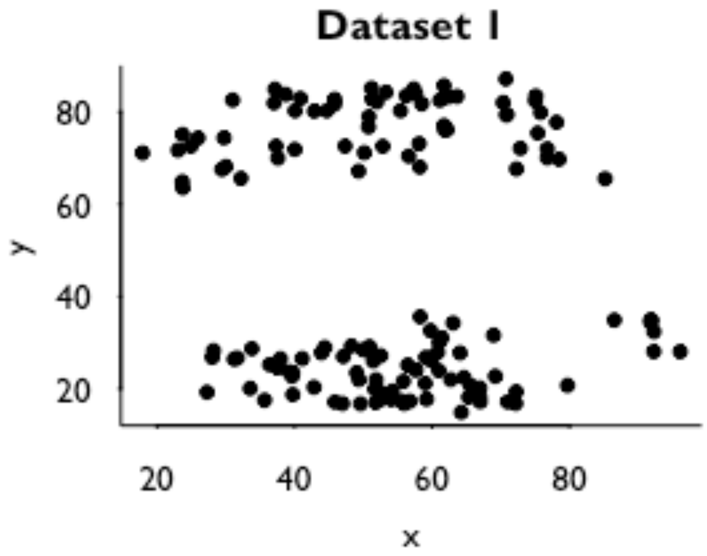
60.98150	23.93430	62.44474	35.4150009	55.06712	29.011536	63.0769	29.1026
59.96706	26.80258	72.46241	27.6242638	61.61478	29.994390	64.1026	25.2564
62.70745	21.99551	69.91943	29.3110924	68.54314	35.657833	64.3590	24.1026
69.19524	22.70979	77.76286	23.4913731	77.70611	20.304260	74.3590	22.9487
65.59236	20.69417	72.77841	26.9005356	68.45305	13.035528	71.2821	22.9487
64.61710	22.44315	65.19832	33.0271721	68.25721	12.384639	67.9487	22.1795
59.19116	17.76790	69.45986	30.1970310	70.25548	13.250385	65.8974	20.2564
55.83963	21.61857	59.31506	12.3543139	65.04433	11.000841	63.0769	19.1026
59.00076	21.17101	51.48771	20.8436108	60.09225	11.872111	61.2821	19.1026
56.45937	25.14456	57.11245	15.3217850	52.99203	9.906669	58.7179	18.3333
52.35810	19.41705	50.22442	19.1709690	50.14463	12.211543	55.1282	18.3333
51.81166	17.03349	50.43386	18.5348701	46.50861	11.207134	52.3077	18.3333
54.22285	19.50367	56.38031	14.7884598	43.80703	11.318945	49.7436	17.5641
49.01876	23.48927	54.86351	14.0096121	57.81786	10.945143	47.4359	16.0256
35.65358	17.51451	45.49986	3.9854262	50.94049	9.691547	44.8718	13.7180
46.96579	16.78920	49.93674	21.5900974	63.49732	11.914069	48.7179	14.8718
49.58424	16.75792	53.80987	17.7664146	50.01648	11.933852	51.2821	14.8718
51.85630	21.96908	59.85268	9.0399504	58.63676	11.974721	54.1026	14.8718
55.95289	16.78976	48.17566	19.5212154	54.73029	11.412883	56.1538	14.1026
51.66379	19.92513	50.74172	17.8384056	65.87555	11.732436	52.0513	12.5641
45.85967	17.12890	58.45976	14.1486626	57.06098	9.920561	48.7179	11.0256

39.74038	18.74105	50.11460	1.7414618	46.81991	10.494653	47.1795	9.8718
52.81122	17.85499	45.42038	5.3724089	38.35939	13.431323	46.1538	6.0256
56.78910	17.30612	51.35327	0.3038724	47.31541	12.853452	50.5128	9.4872
54.11453	17.67353	53.09171	19.2900295	55.05192	11.949988	53.8462	10.2564
55.57112	17.36958	58.94316	11.6128779	50.51596	9.765592	57.4359	10.2564
65.30647	18.19736	59.31099	10.5721359	49.67741	10.383133	60.0000	10.6410
66.99075	17.22287	68.22220	1.1338804	67.28066	14.128652	64.1026	10.6410
72.15672	16.81428	62.51177	7.4912324	66.17302	12.037917	66.9231	10.6410
70.98058	17.15575	69.38006	1.4881324	61.08854	10.084535	71.2821	10.6410
72.22208	19.36928	82.80025	18.5293770	66.05309	13.380226	74.3590	10.6410
79.70016	20.74022	85.16033	15.0183697	72.66999	15.234226	78.2051	10.6410
64.24916	14.91396	67.09585	0.6014909	61.50347	10.828415	67.9487	8.7180
66.94329	20.18913	62.30775	5.7499747	68.99503	13.994310	68.4615	5.2564
66.94167	18.24351	59.97041	7.9133105	78.24992	17.883242	68.2051	2.9487
39.64827	22.73137	39.91164	32.1440239	36.48198	15.162760	37.6923	25.7692
37.94978	26.52175	39.59650	32.1361313	50.96775	29.679775	39.4872	25.3846
86.50439	34.89402	95.59341	33.2341309	91.19106	46.674343	91.2821	41.5385
50.81549	78.95084	52.77229	99.6134720	55.86377	85.336487	50.0000	95.7692
51.18347	85.12885	53.17862	98.6038590	49.28059	84.048821	47.9487	95.0000
40.86385	82.92615	37.62705	93.0606766	43.36850	84.332176	44.1026	92.6923

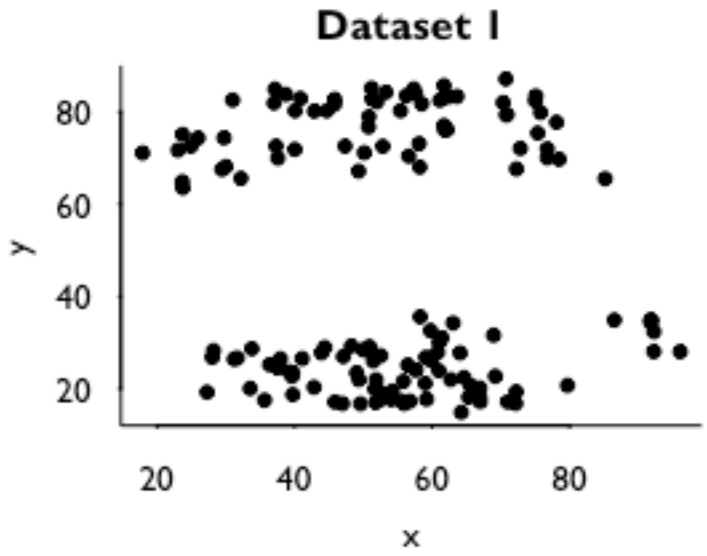
Ok, let's try something else

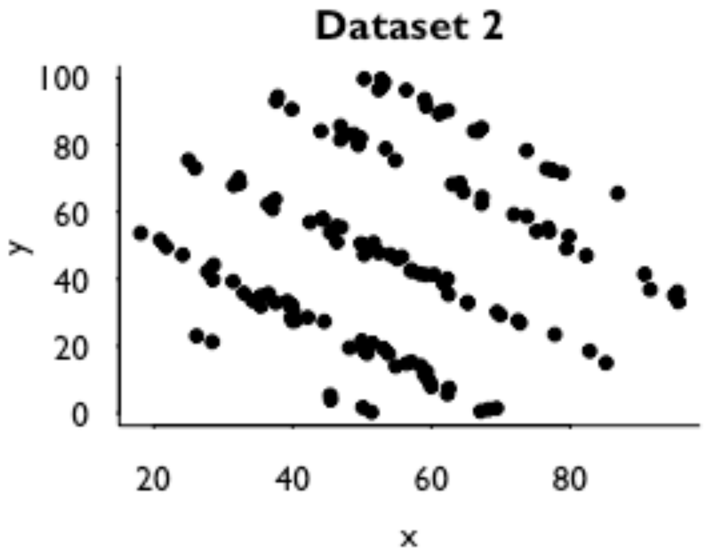
	Dataset 1	*Dataset 2*	*Dataset 3*	*Dataset 4*
N	142	142	142	142
$\overline{x_x}$	54.3	54.3	54.3	54.3
$\overline{x_y}$	47.8	47.8	47.8	47.8
σ_x	16.7	16.7	16.7	16.7
σ_y	26.8	26.8	26.8	26.8
ρ	-0.1	-0.1	-0.1	-0.1
α	53.8	53.8	53.8	53.8
β	-0.1	-0.1	-0.1	-0.1
R^2	.005	.005	.005	.005

How About This?

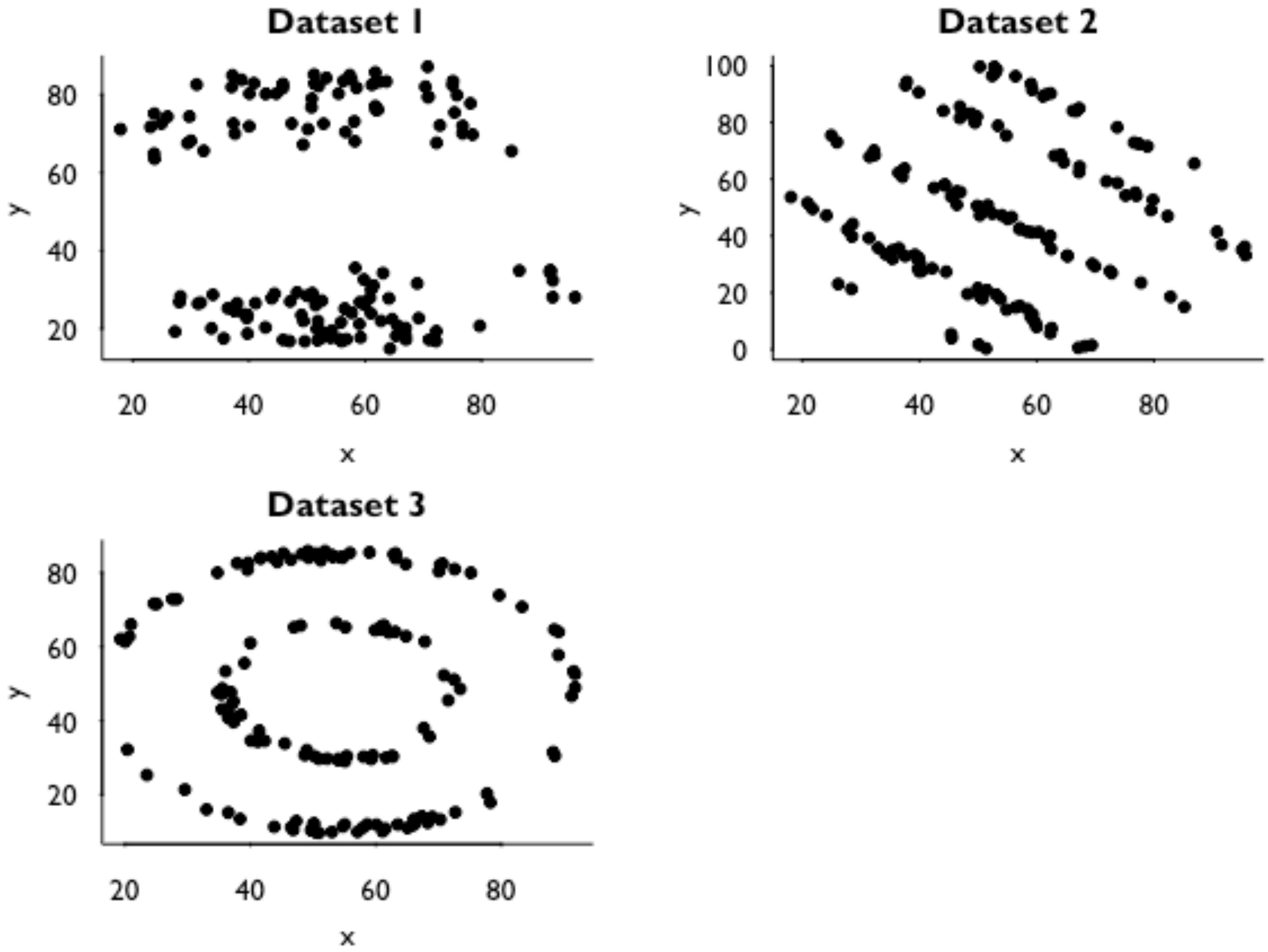


How About This?

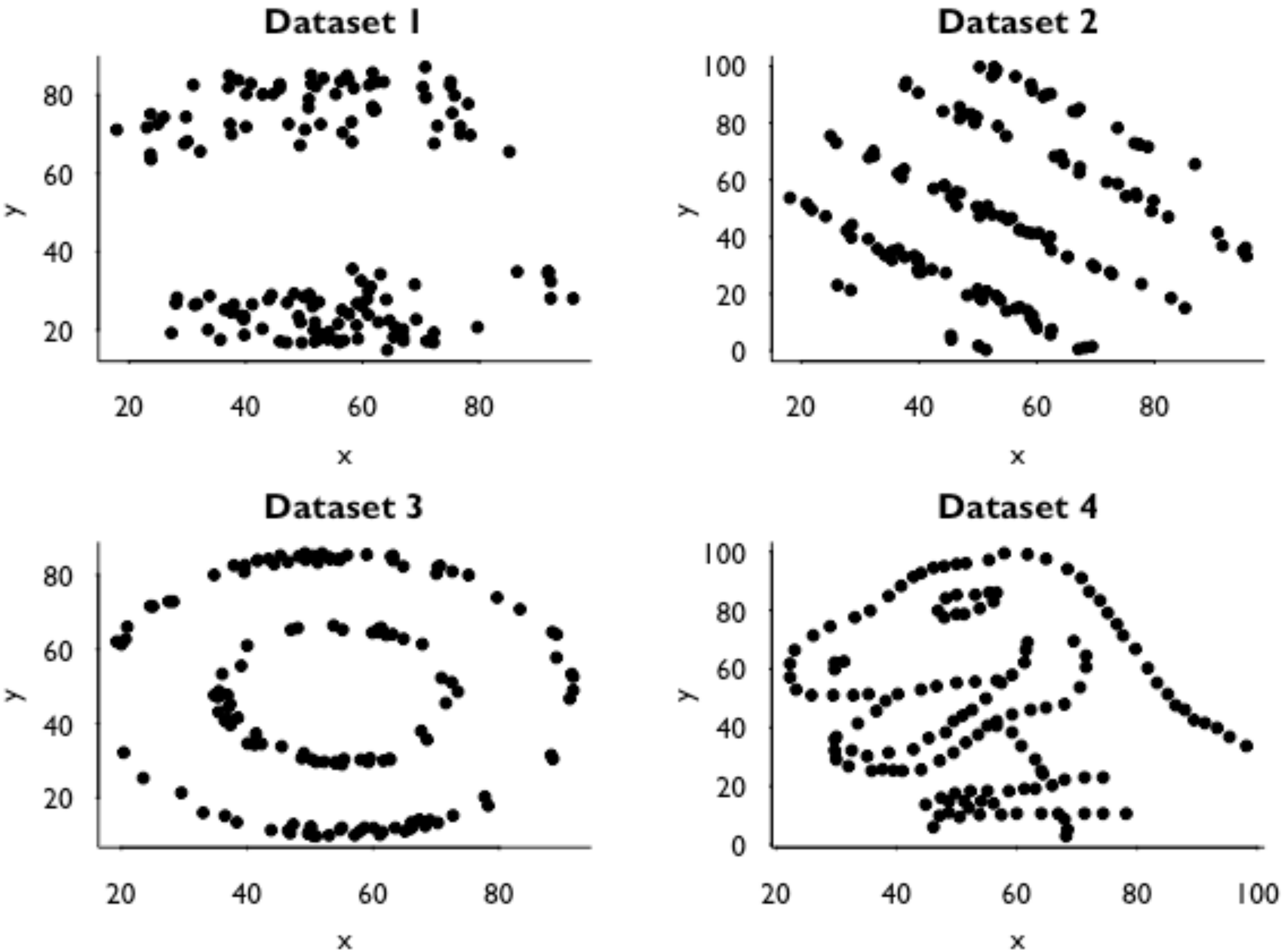




How About This?



How About This?



The Case for Data Visualization

Visualization gives data form or shape, which allows us to see things that are otherwise difficult or impossible to see.

Useful summaries for large, complicated data sets – in fact, the utility of visualization increases with data size.

Little or no assumptions about the nature of the data.

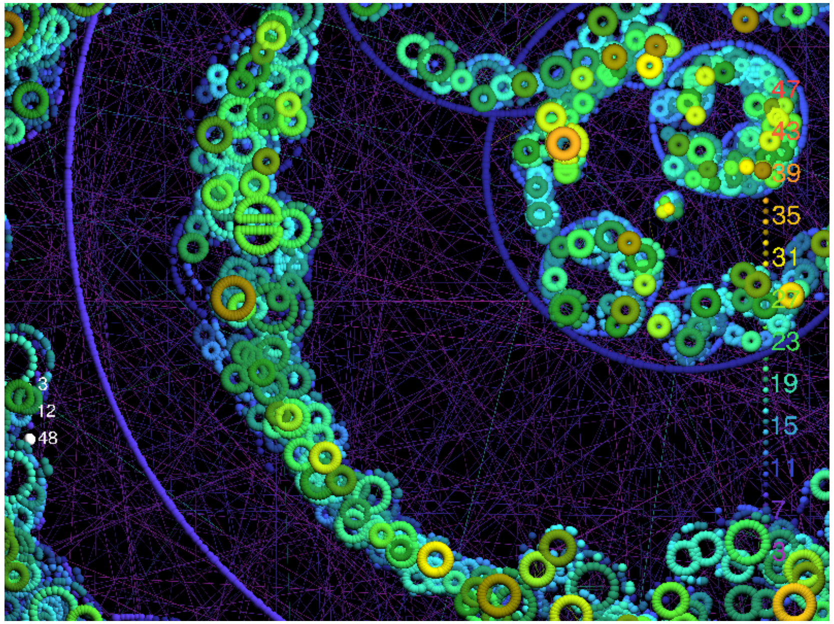
Facilitates interaction between researcher and data – it's a hypothesis generating device.

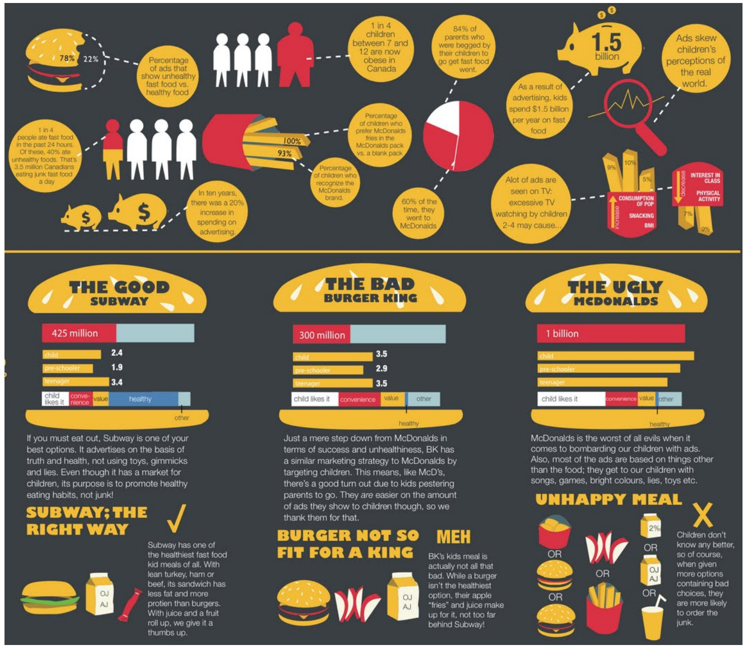
This Course

Some fundamental principles of data visualization

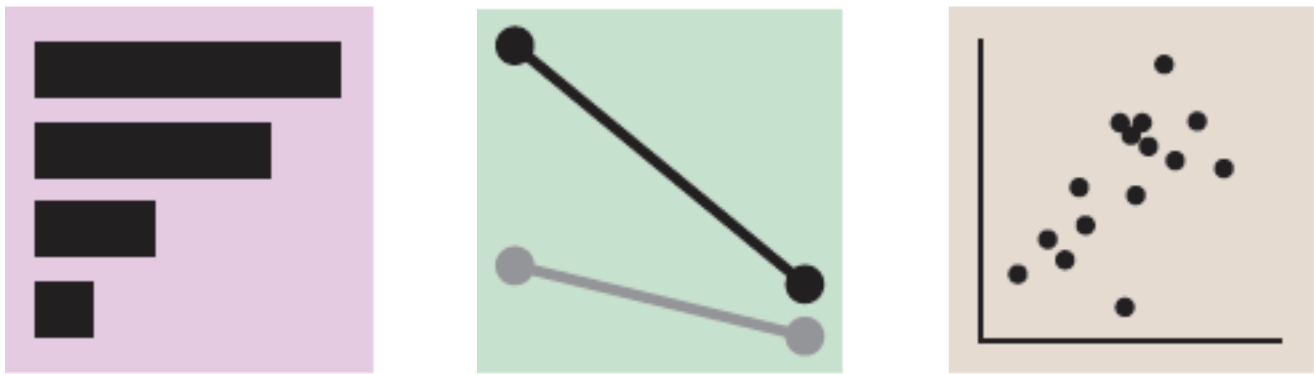
How to do some of it in R

You will probably be disappointed if you expect to see things like this…





Instead I hope to convince you, that…



I know, I know – it's a bit like hoping for this...

And getting this...

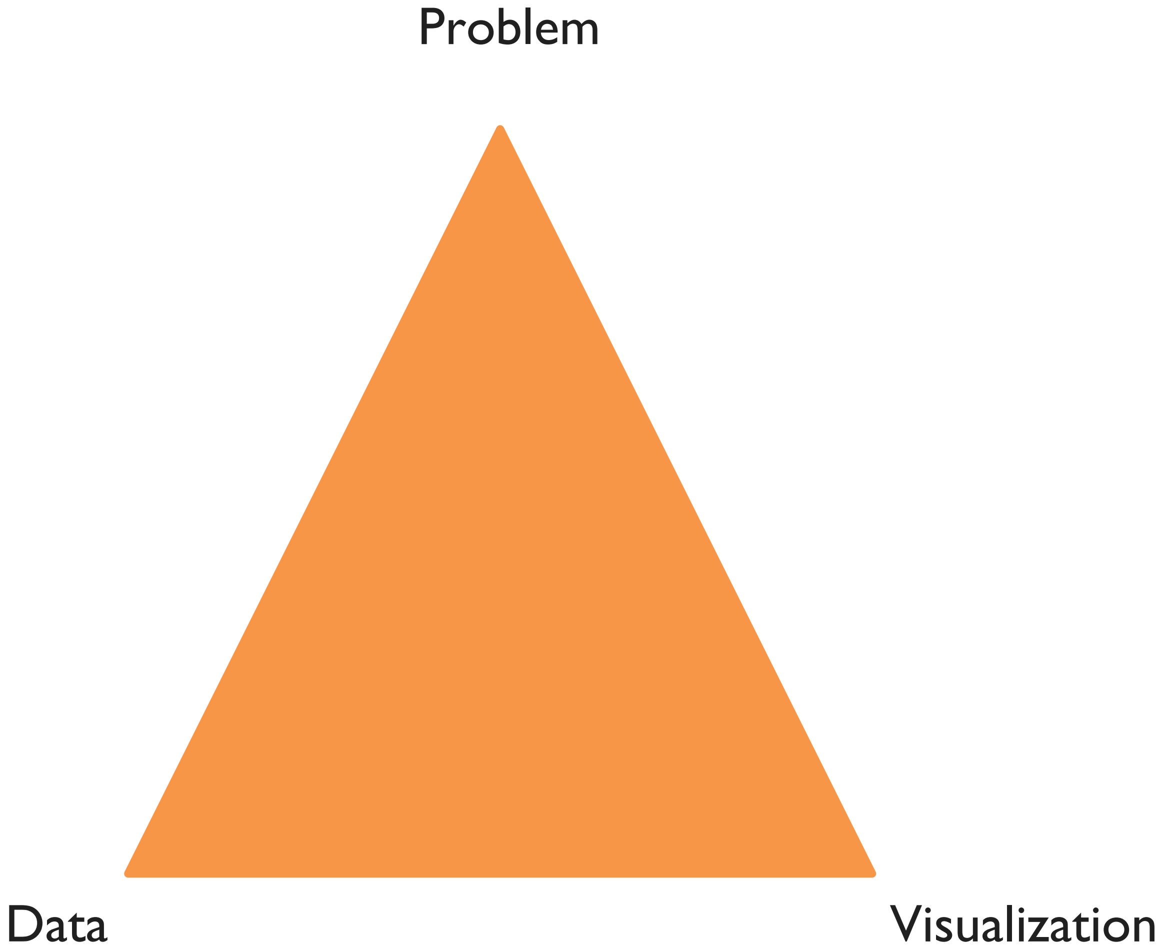
Data Visualization as a Methodology

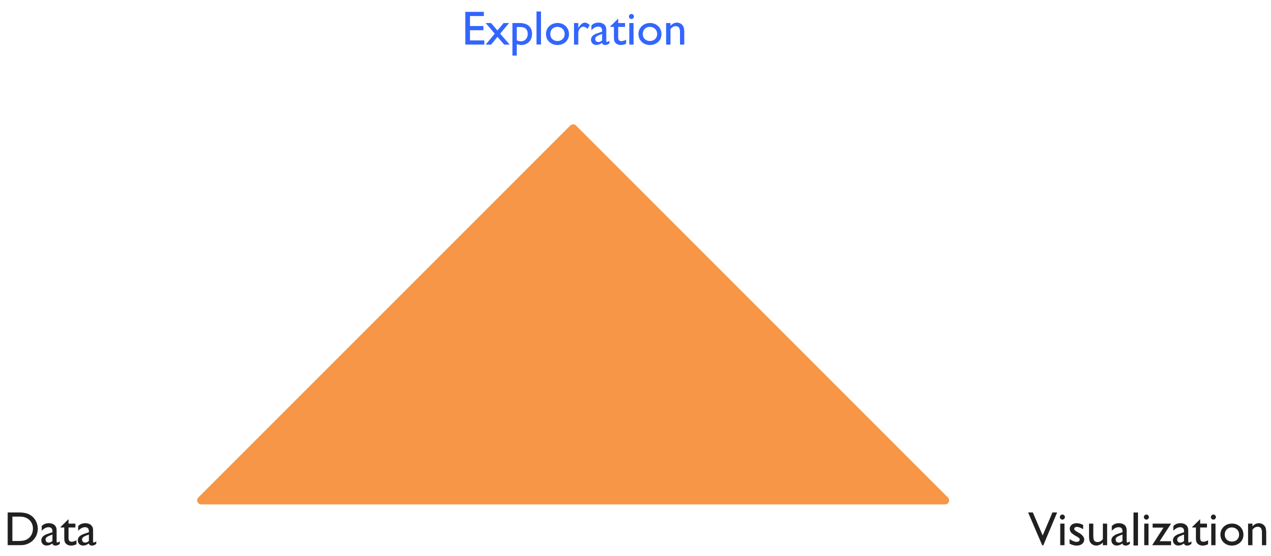
“The critical question is how best to transform the data into something that people can understand for optimal decision making.“ (Ware 2013: 5)

Data visualization is a method for making sense of quantitative information – not to make pictures of data.

Note that this is also more than data visualization in the narrow sense, i.e. the act of encoding quantitative information in visual objects.

Data Visualization as a Methodology

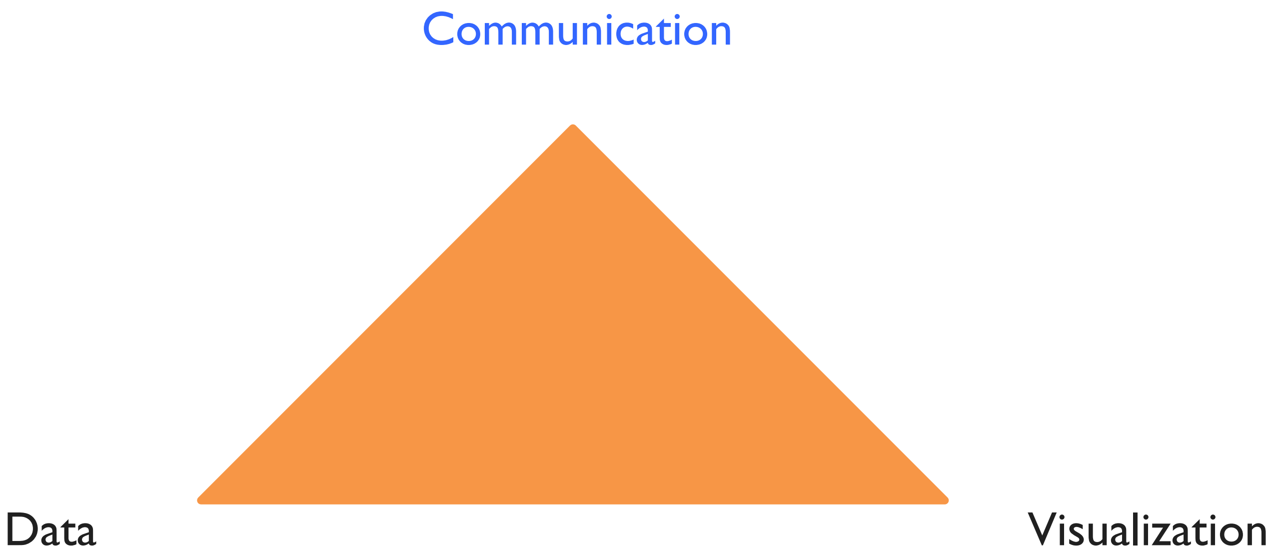




What's in the data?

Get a sense of size and complexity of data.

Explore and interact.



Communicate content of data.

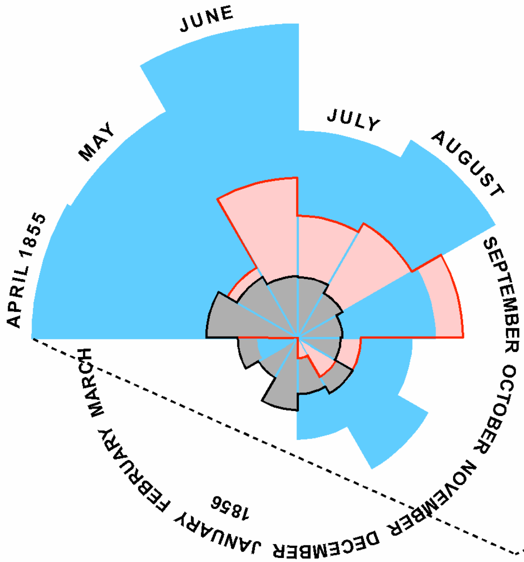
Tell a story with data.

Attract attention and interest.

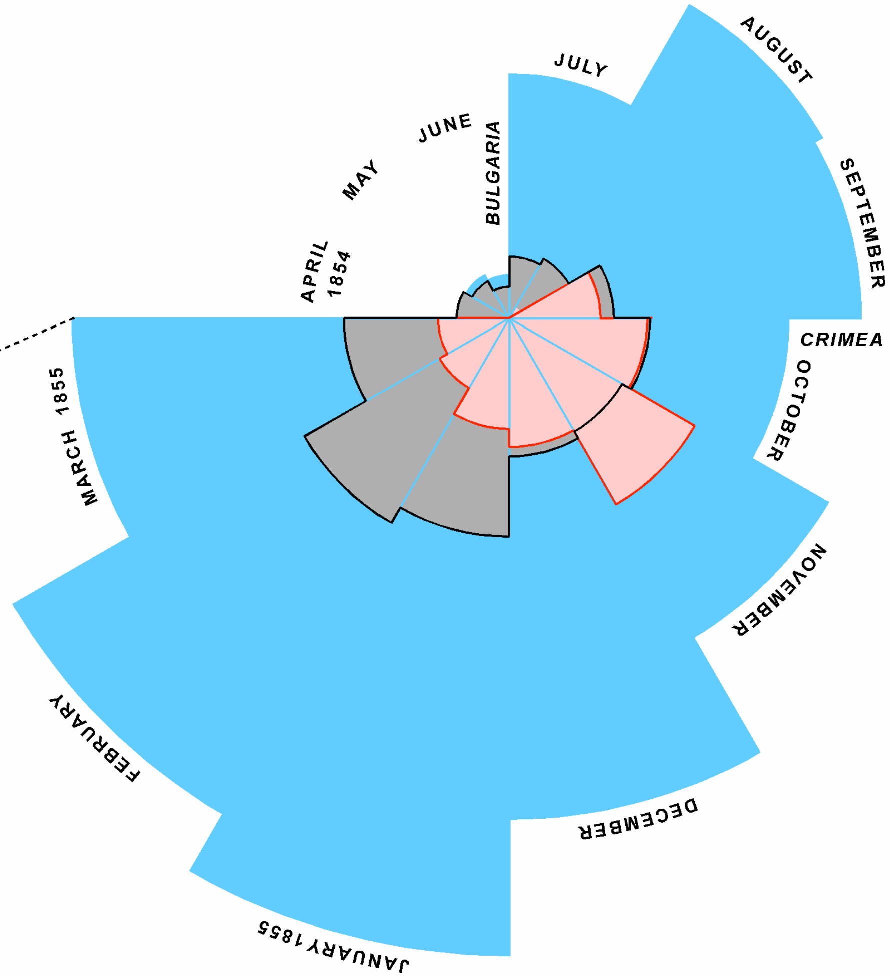
http://www.Florence-Nightingale-Avenging-Angel.co.uk/Coxcomb.htm

DIAGRAM OF THE CAUSES OF MORTALITY
IN THE ARMY IN THE EAST.

2.
APRIL 1855 TO MARCH 1856.



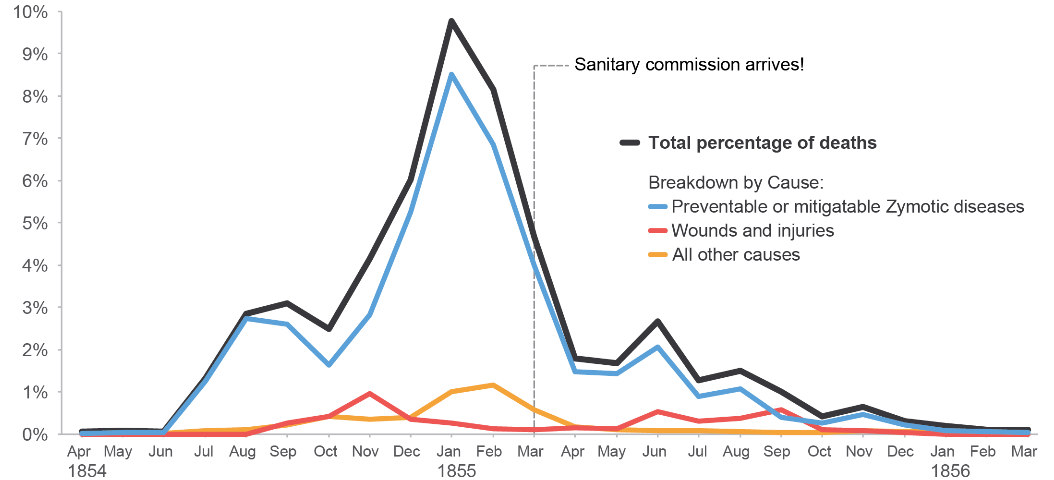
1.
APRIL 1854 TO MARCH 1855.



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex
The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic Diseases, the red wedges measured from the centre the deaths from wounds, & the black wedges measured from the centre the deaths from all other causes
The black line across the red triangle in Nov^r 1854 marks the boundary of the deaths from all other causes during the month
In October 1854, & April 1855, the black area coincides with the red, in January & February 1856, the blue coincides with the black
The entire areas may be compared by following the blue, the red & the black lines enclosing them

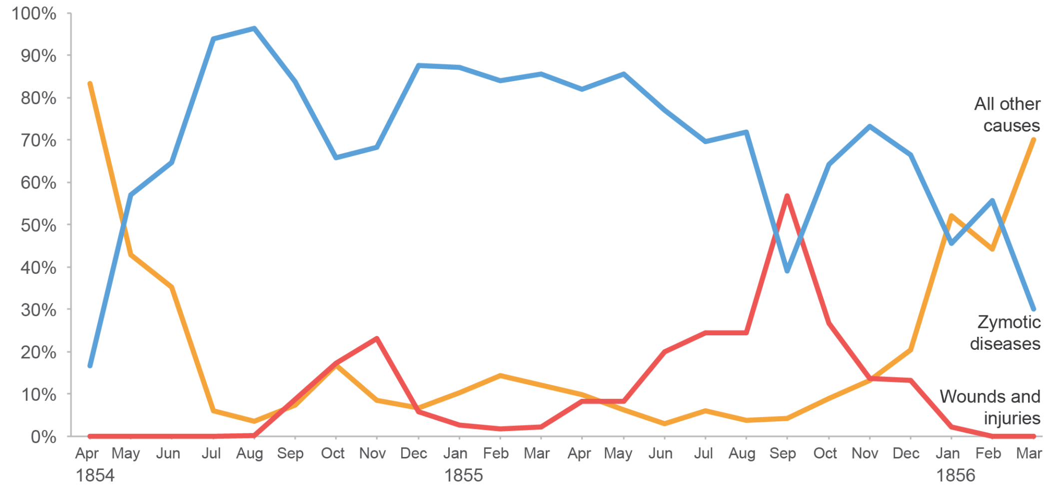
Most Deaths Among Soliders in the East are Preventable

Percentage of Active Soldiers Who Died per Year and Cause



Notice in the graph above that most deaths in the war effort (black line represents total deaths) were caused by preventable and mitigatable Zymotic diseases (blue line, which closely hugs the black line).

Percentage of Total Deaths by Cause



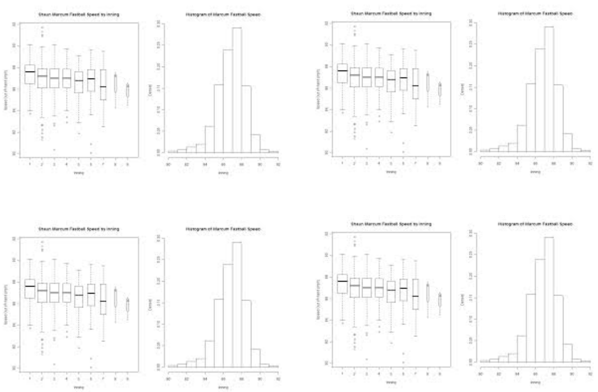
Notice that wounds and injuries (red line) at one point in time only–September 1855–accounted for a greater percentage of deaths than Zymotic diseases and otherwise accounted for relatively few.

Exploratory Visualization

„forces us to notice what we never expected to see“ (Tukey 1977: vi)

Mostly for ourselves in the course of the research process.

Many, quick and dirty, and rather unattractive graphs

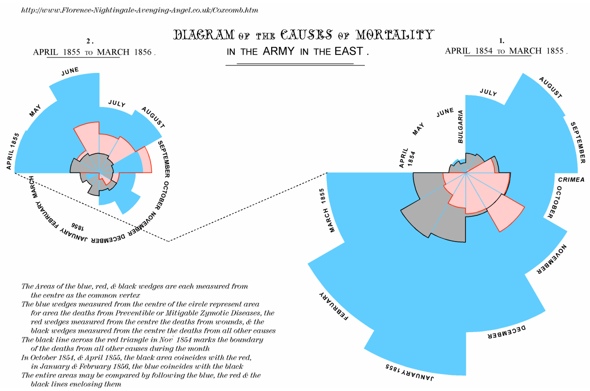


Explanatory Visualization

„forces readers to see the information the designer wanted to convey“ (Kosslyn 1994: 271).

Mostly for others after the research is completed.

Few, carefully crafted, and attractive graphs.



The Fundamental Principles of Analytic Design

1. Look at and Show: Comparisons, Contrasts, Differences

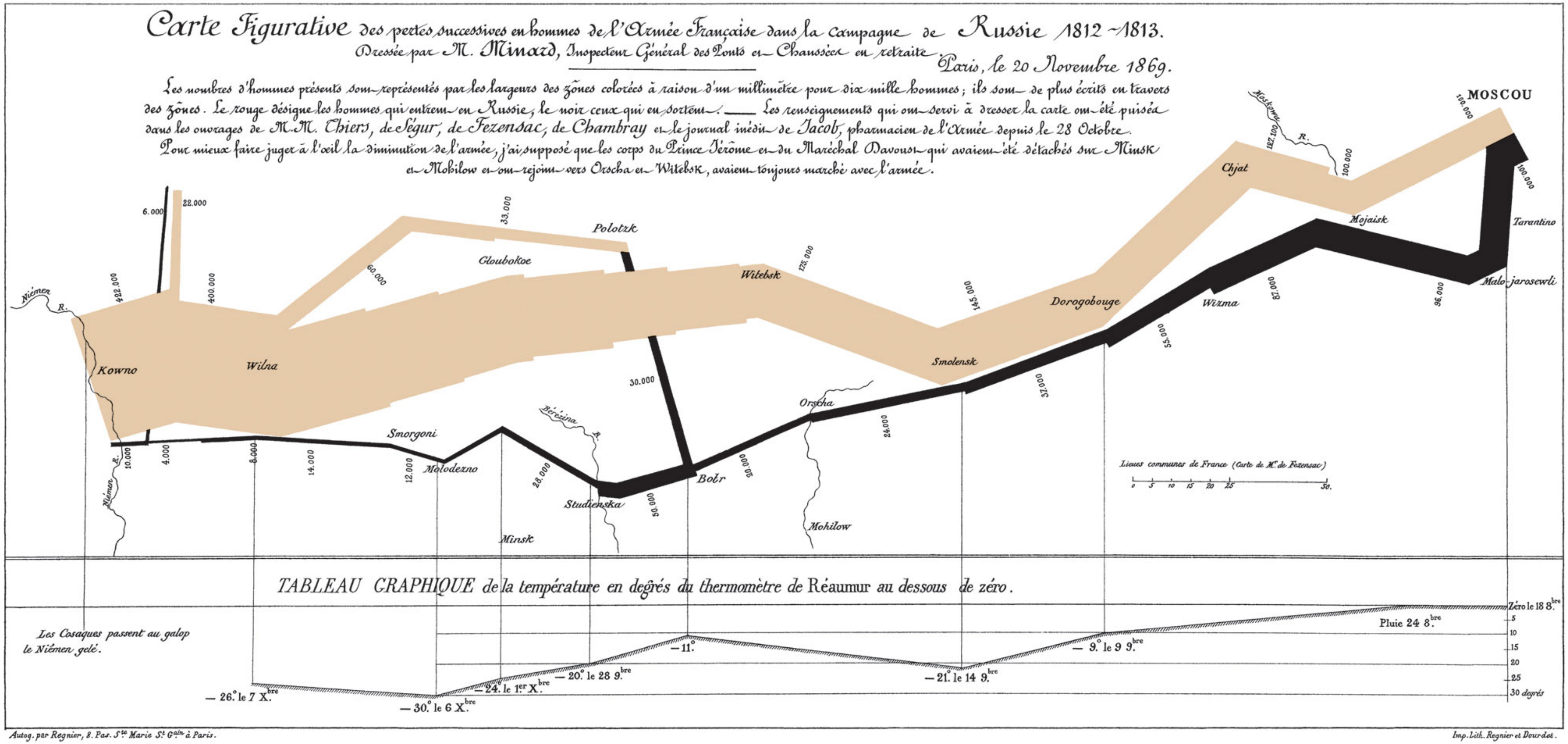
2. Look at and Show: Causality, Mechanism, Explanation, Systematic Structure

3. Look at and Show: Multivariate Data; that is, use more than 1 or 2 variables

4. Completely integrate words, numbers, images, diagrams

5. Thoroughly describe the evidence. Provide a detailed title, indicate the authors and sponsors, document the data sources, show complete measurement scales, point out relevant issues.

6. Analytical visualizations ultimately stand or fall depending on the quality, relevance, and integrity of their content. (What is the problem you want to solve?)


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 Mohilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Niémen R.
Kowno
Wilna
Gloubokoe
Polotzk
Witebsk
Smolensk
Dorogobouge
Wizma
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Moskowa R.
Smorgoni
Molodezno
Bérézina R.
Studienska
Minsk
Bobr
Orscha
Mohilow
422.000
400.000
22.000
6.000
60.000
33.000
175.000
145.000
127.100
100.000
30.000
10.000
4.000
8.000
14.000
12.000
28.000
50.000
20.000
24.000
37.000
55.000
87.000
96.000
Lieues communes de France (Carte de M. de Fezensac)
0 5 10 15 20 25 50.
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 1.er 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
5
10
15
20
25
30 degrés
Autog. par Regnier, 8. Pas. S.te Marie S.t G.ain à Paris.
Imp. Lith. Regnier et Dourdet.

Graphical Perception

Graphical Perception

When a graph is constructed, quantitative and categorical information is encoded, chiefly through position, shape, size, symbols, and color.

23, 34, 54 → 

When a person looks at a graph, the information is visually decoded by the person's visual system.

 → 

Graphical Perception

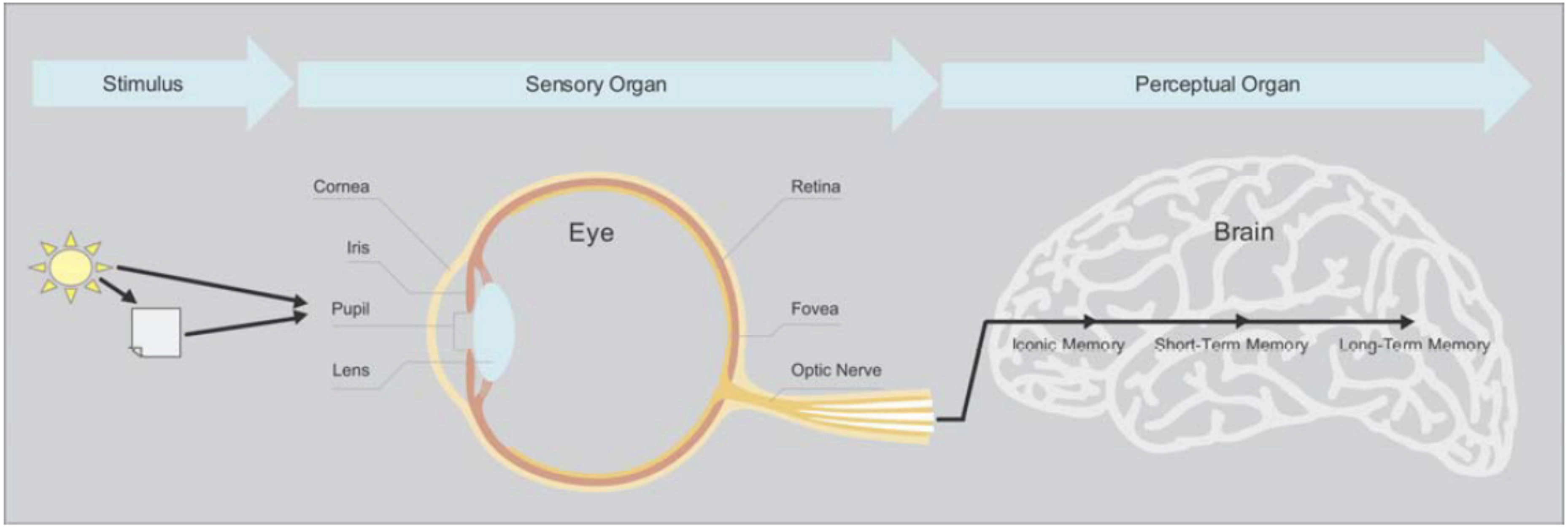
“If we can understand how perception works, our knowledge can be translated into rules for displaying information.

Following perception-based rules, we can present our data in such a way that the important and informative patterns stand out.

If we disobey the rules, our data will be incomprehensible or misleading.”

(C. Ware 2004: xxi)

How Perception Works



Iconic Memory (Visual Sensory Register):	Pre-attentive processing (unconscious and automatic) of visual attributes (colors, shapes, etc.)
Short Term or Working Memory:	Attentive processing of "chunks" of information
Long Term Memory:	Storing and recognizing familiar patterns

Attributes of Pre-attentive Processing

756395068473

658663037576

860372658602

846589107830

How many 3s are there?

Attributes of Pre-attentive Processing

756395068473

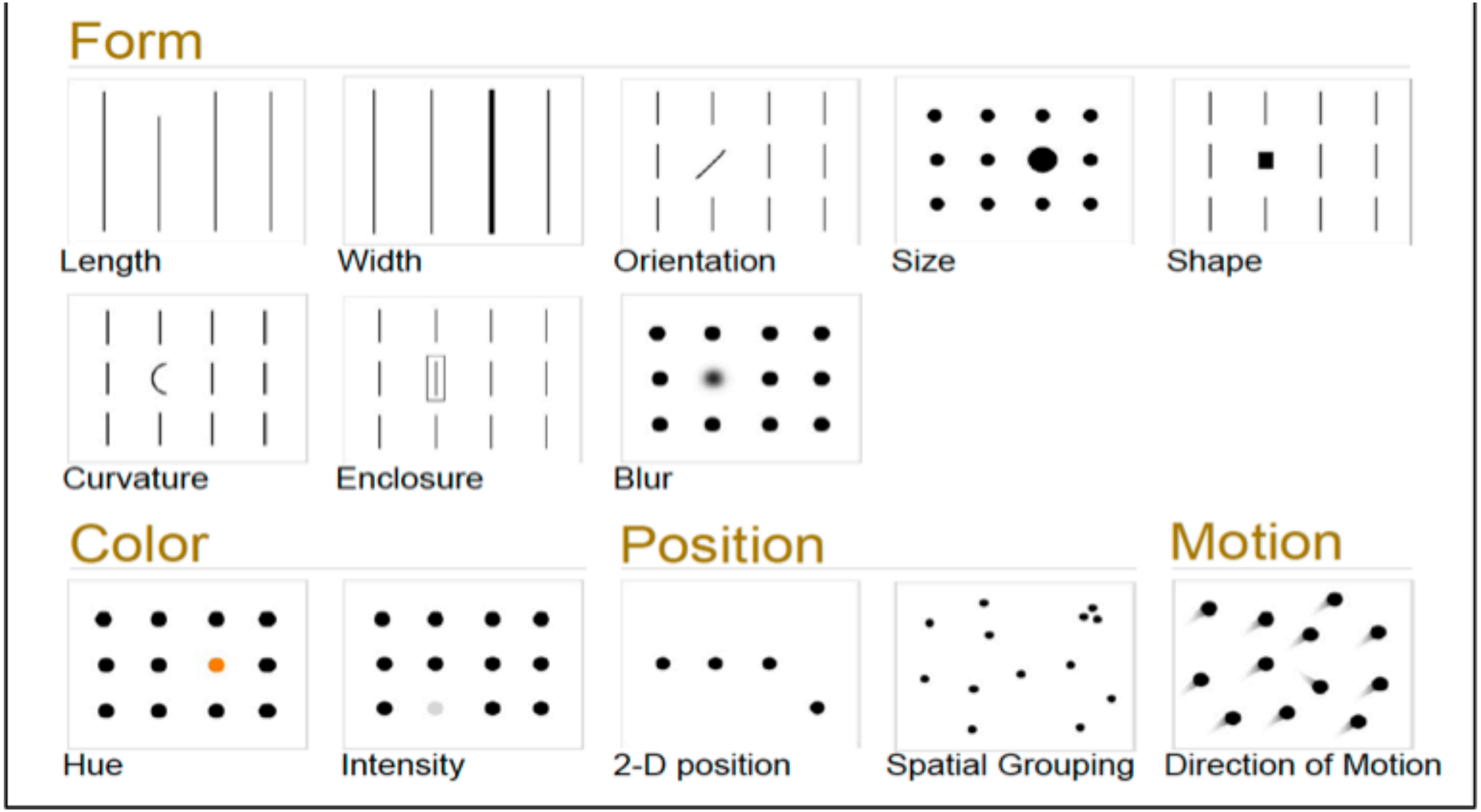
658663037576

860372658602

846589107830

How many 3s are there?

Attributes of Pre-attentive Processing



Attributes of Pre-attentive Processing

We can use attributes of pre-attentive processing to

1. Choose visual encodings of data (choice of graphical formats)

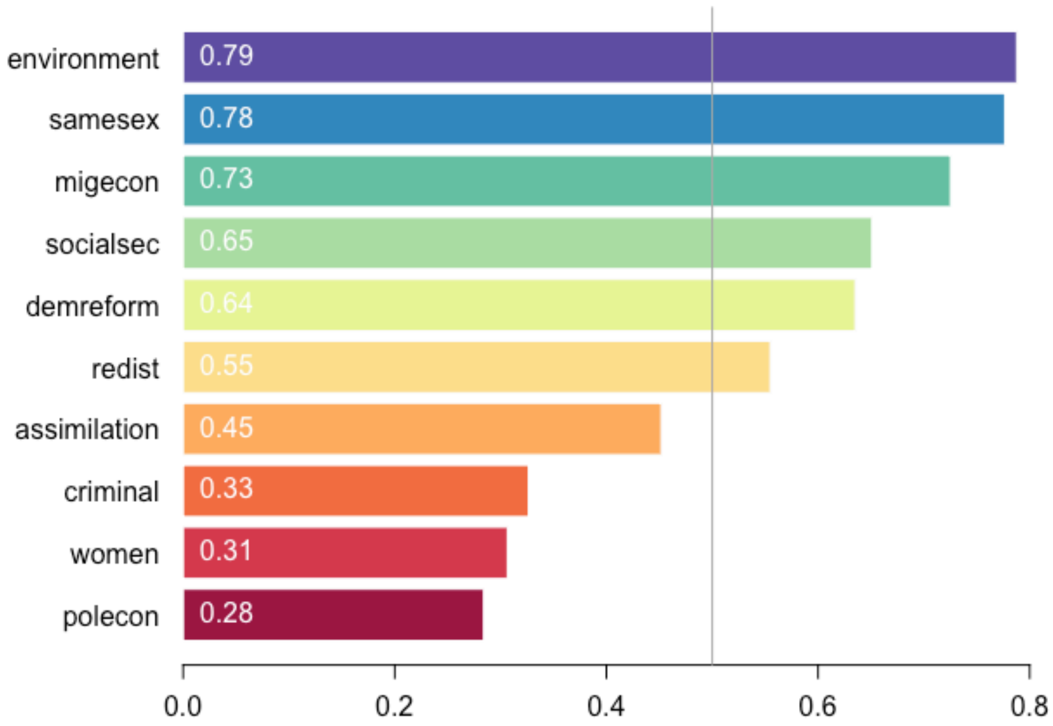
2. Create a visual hierarchy (design choice)

Attributes of Pre-attentive Processing

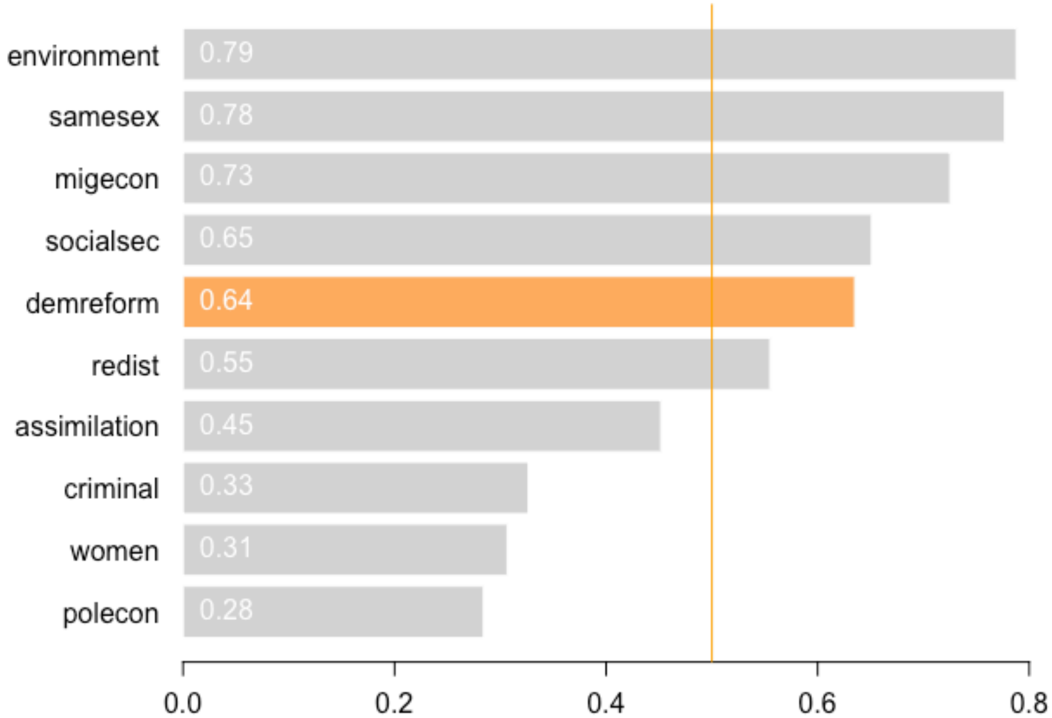
Attributes of Pre-attentive Processing

Pre-attentive attributes become less distinct as the variety of distractors increases.

Example


environment
0.79
samesex
0.78
migecon
0.73
socialsec
0.65
demreform
0.64
redist
0.55
assimilation
0.45
criminal
0.33
women
0.31
polecon
0.28
0.0
0.2
0.4
0.6
0.8

Example

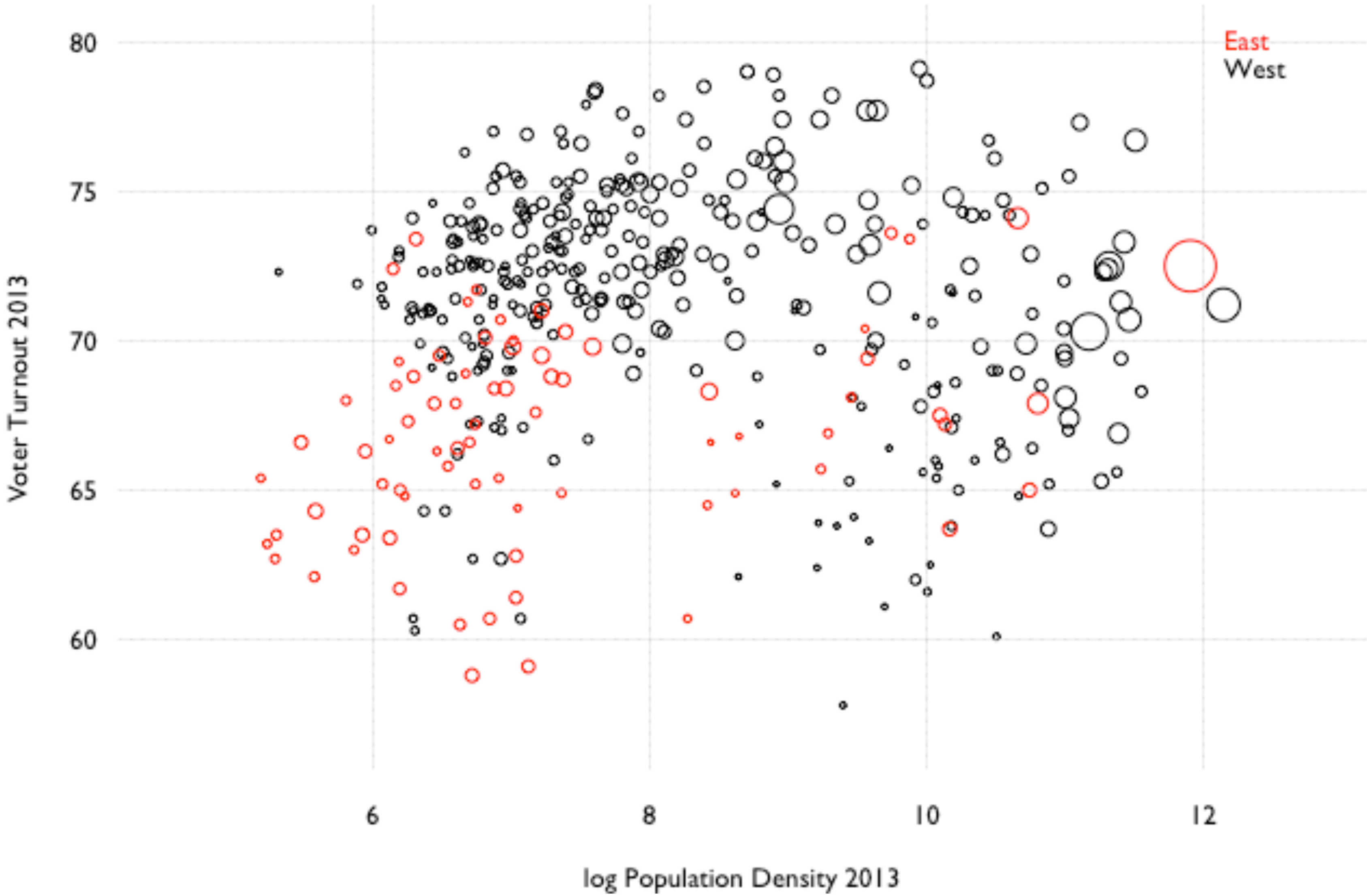


environment
0.79
samesex
0.78
migecon
0.73
socialsec
0.65
demreform
0.64
redist
0.55
assimilation
0.45
criminal
0.33
women
0.31
polecon
0.28
0.0
0.2
0.4
0.6
0.8

Attributes of Pre-attentive Processing

How much bigger is the second circle?

Example

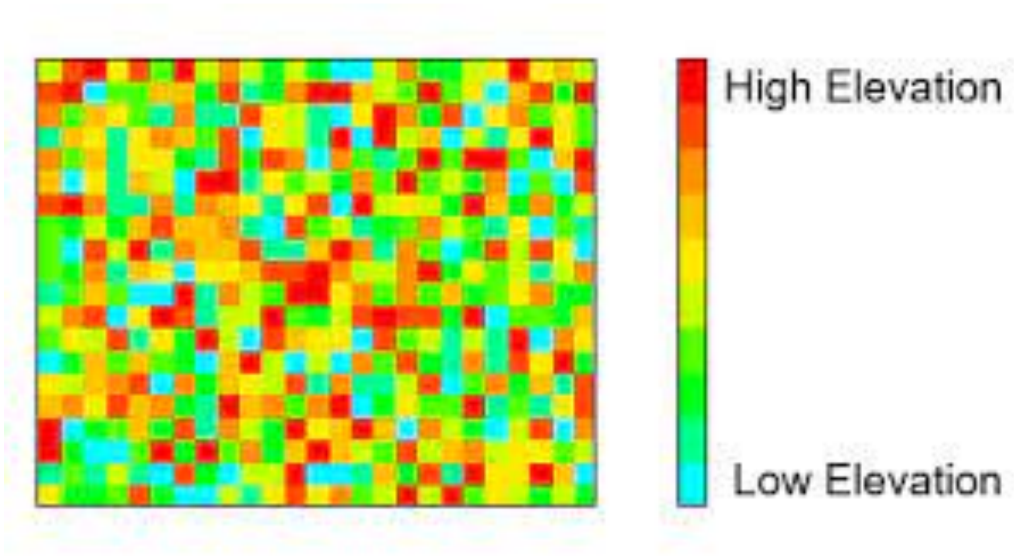


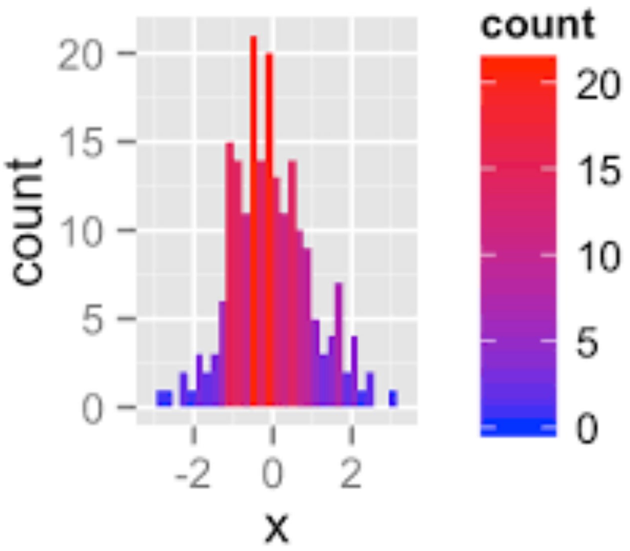
80
75
70
65
60
East
West
Voter Turnout 2013
6
8
10
12
log Population Density 2013

Attributes of Pre-attentive Processing

Bring the colors in an order from small to large!

Example


High Elevation
Low Elevation


count
20
15
10
5
0
-2
0
2
x
count
20
15
10
5
0

Visual attributes are not all created equal…

Attribute	*Quantitatively Perceived?*
2-D Position	Yes
Length	Yes
Width	Limited
Size	Limited
Color Intensity	Limited
Orientation	No
Shape	No
Enclosure	No
Color Hue	No

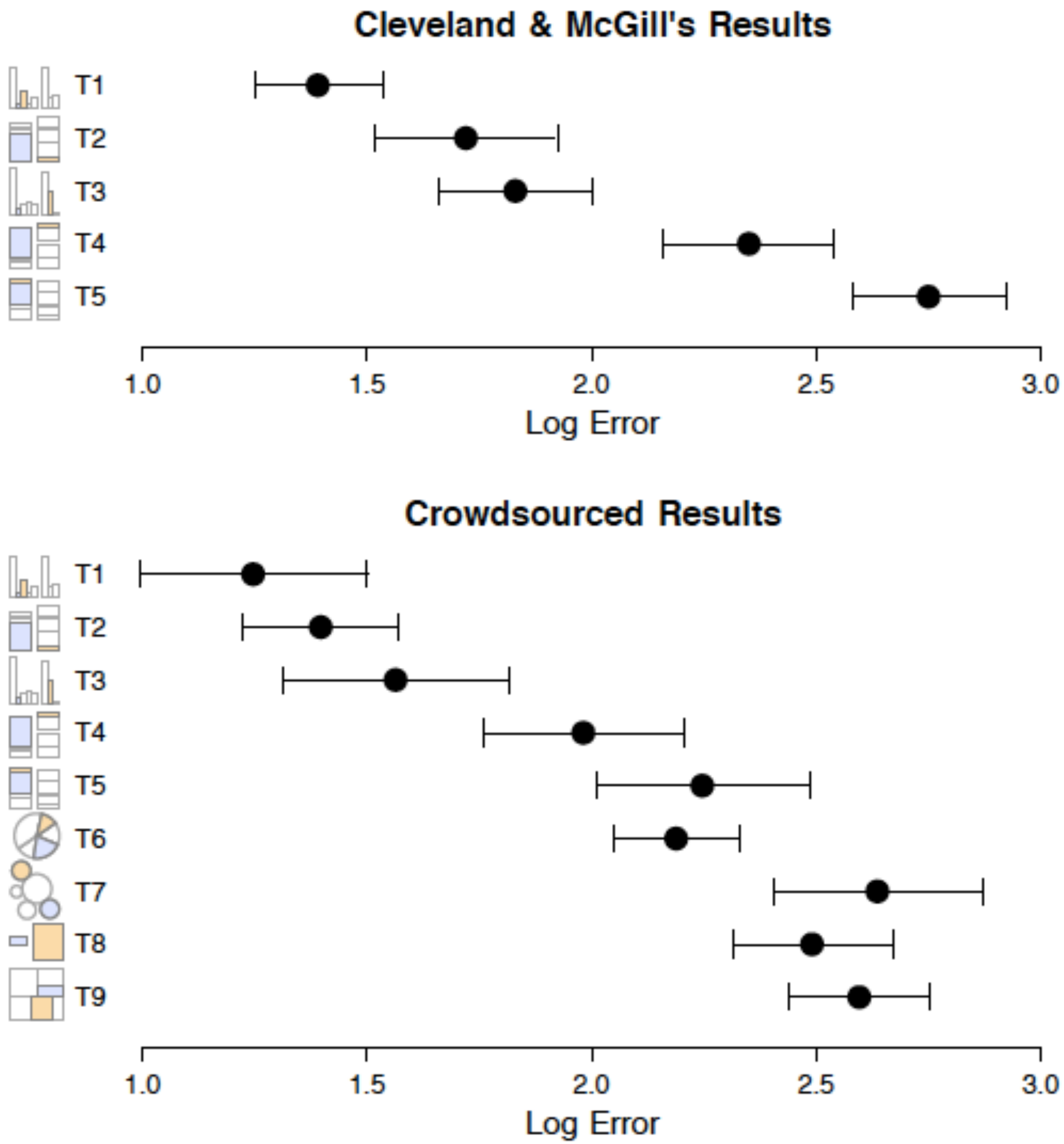
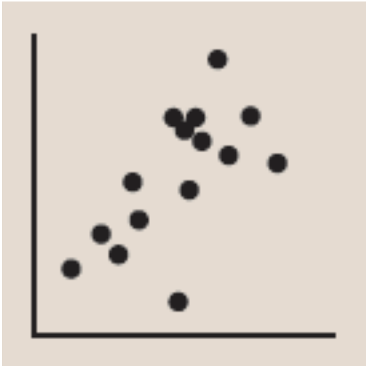
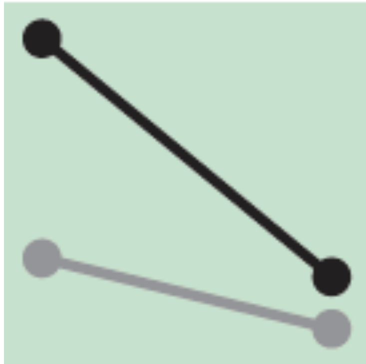


Figure 4: Proportional judgment results (Exp. 1A & B). Top: Cleveland & McGill's [7] lab study. Bottom: MTurk studies. Error bars indicate 95% confidence intervals.

(Heer & Bostock 2010)

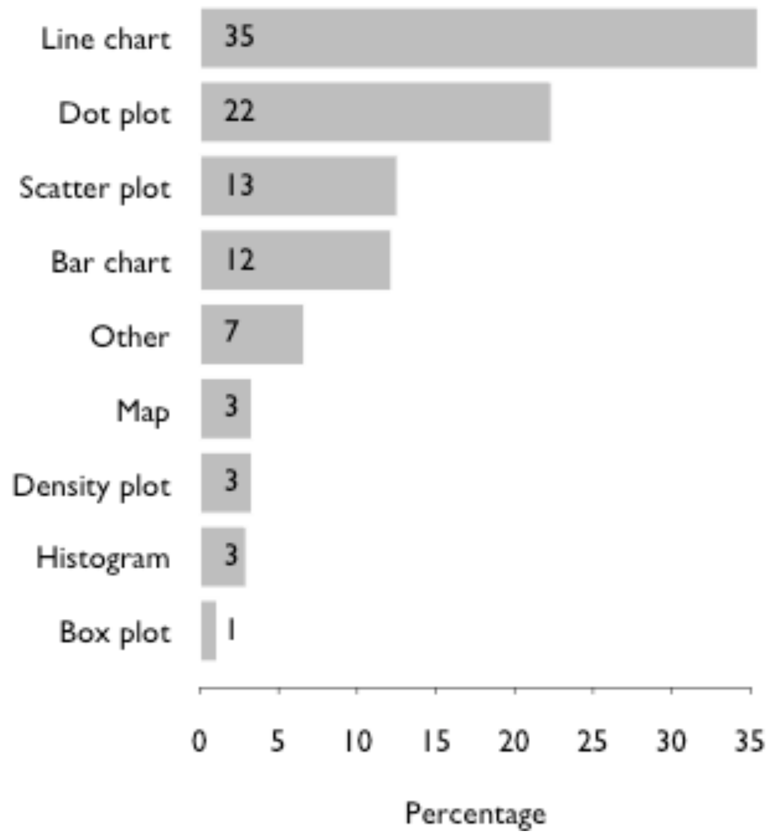
Now you know why…



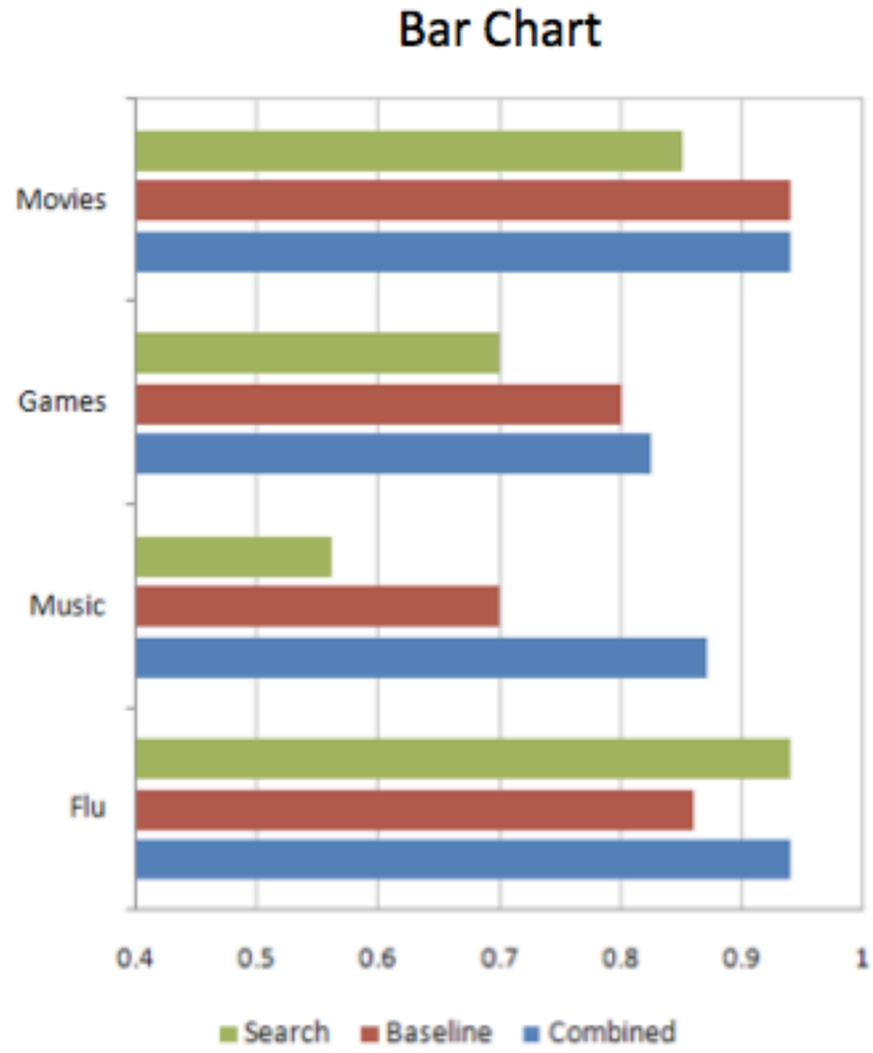


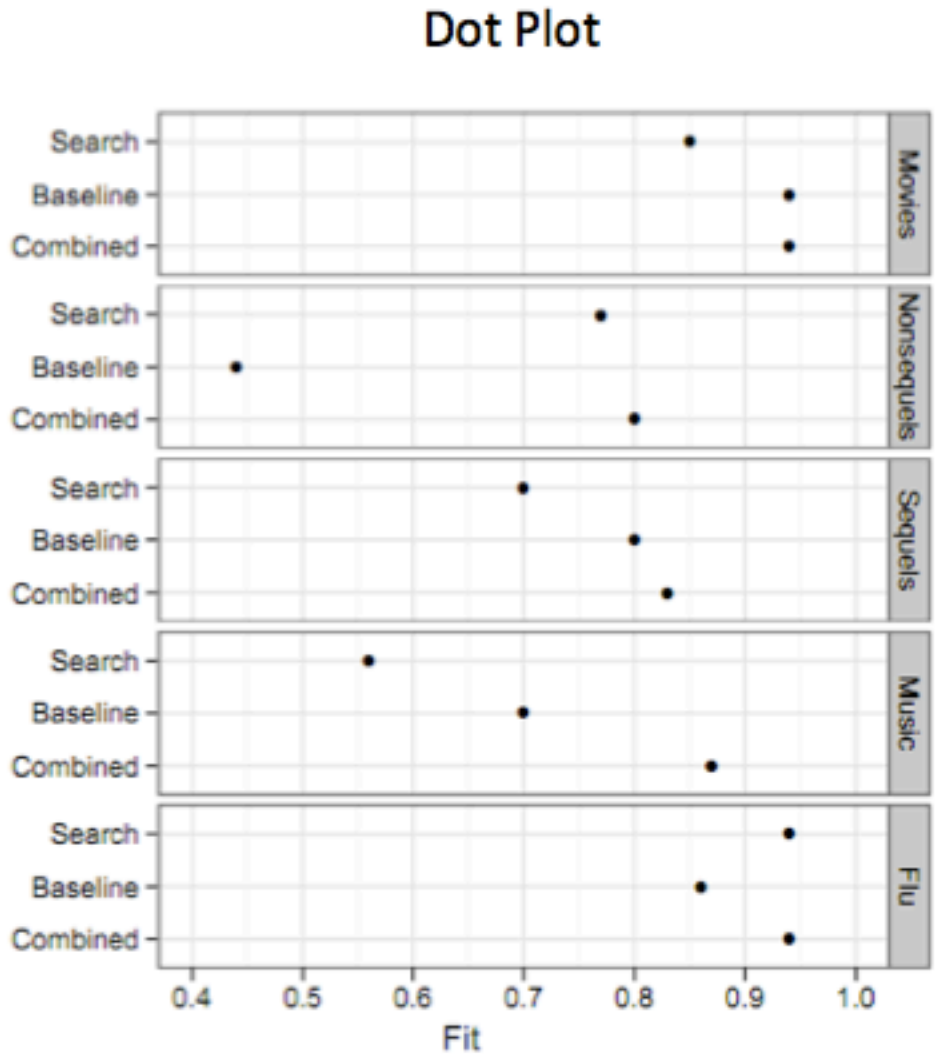
Dot Chart/Scatter Plot:	2-D position of visual objects
Line Chart:	2-D position, connected to give shape to a series of values
Bar Chart:	Length and 2-D position

Graphical Format Used in AJPS 2003-2018



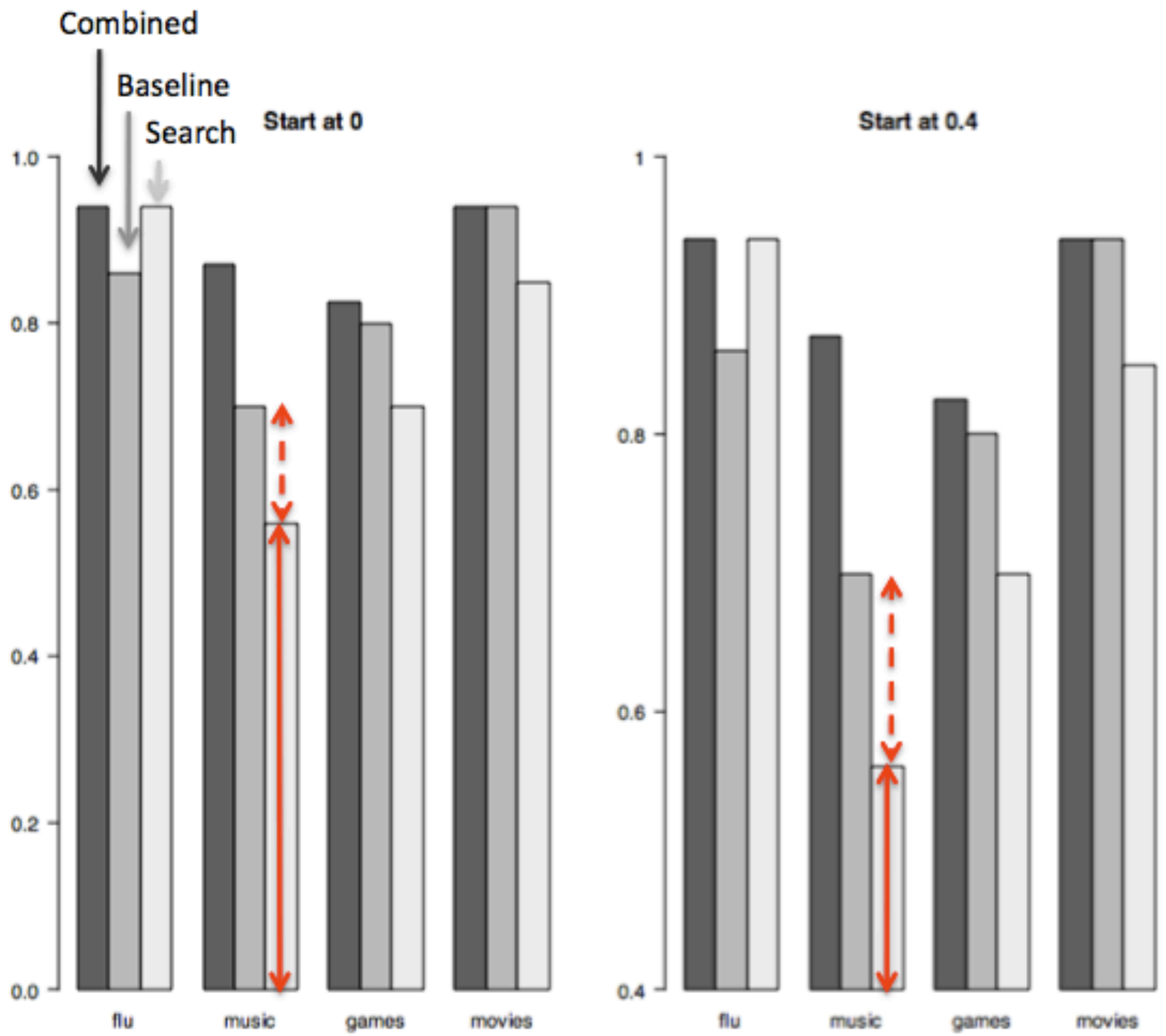
Trade-Offs and Comparative Advantages



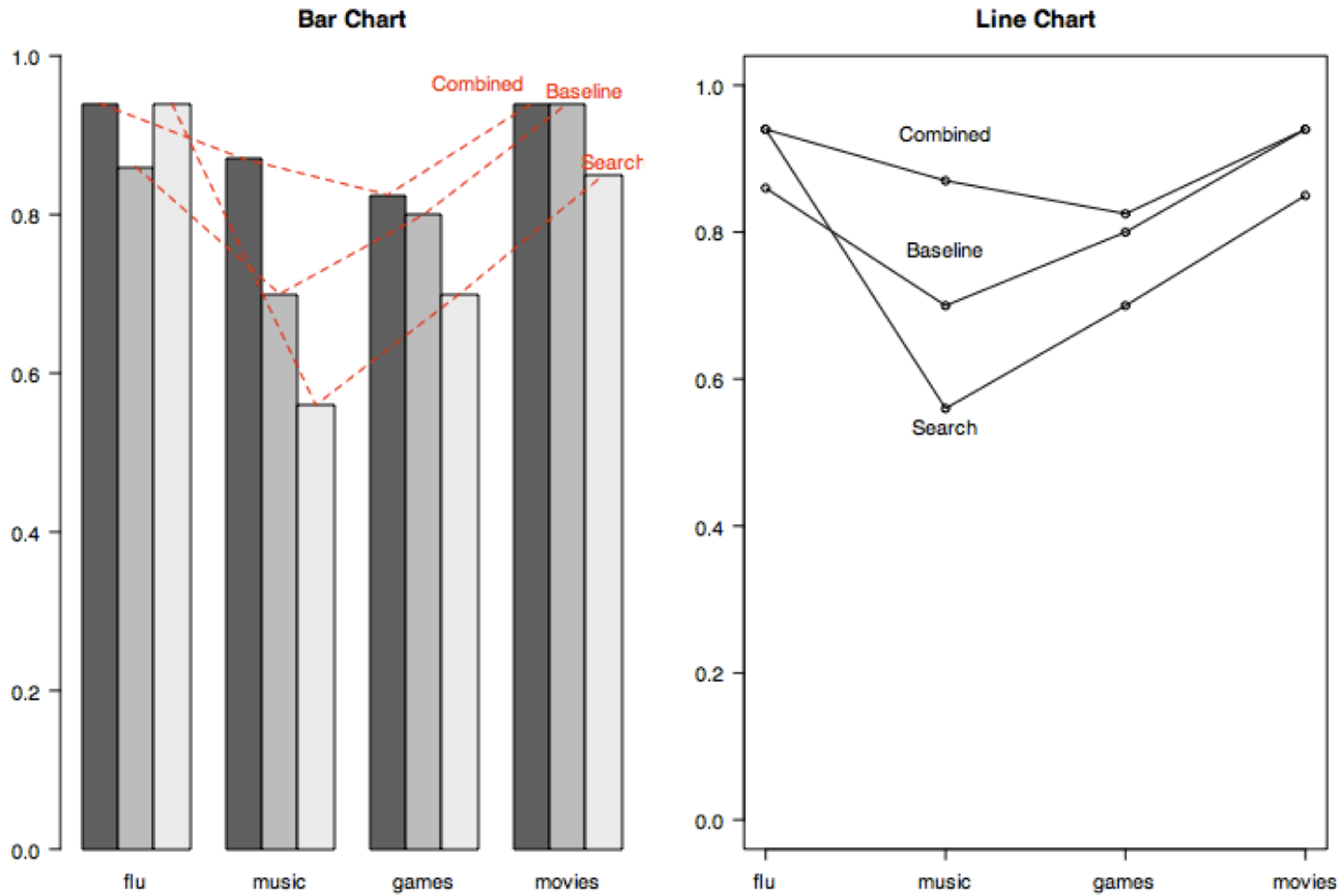


Source: Junkcharts.

Trade-Offs and Comparative Advantages



Trade-Offs and Comparative Advantages



The Zen of Visualization Design

Above else show the data.

Maximize the data-ink ratio.

Erase non-data-ink.

Erase redundant data-ink.

Revise and edit.

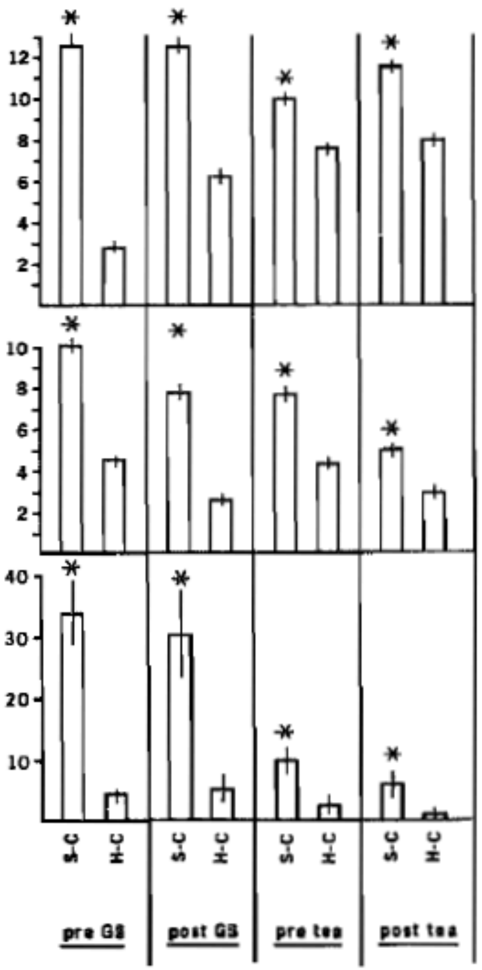
The Zen of Visualization Design

$$\text{Data-ink Ratio} = \frac{\text{data ink}}{\text{total ink used to print the graphic}}$$

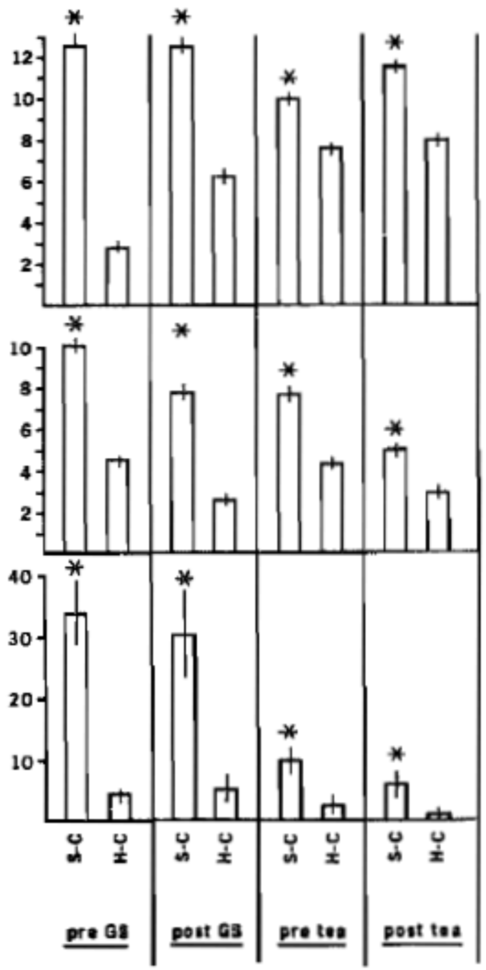
= proportion of ink devoted to the non-redundant display of data-information

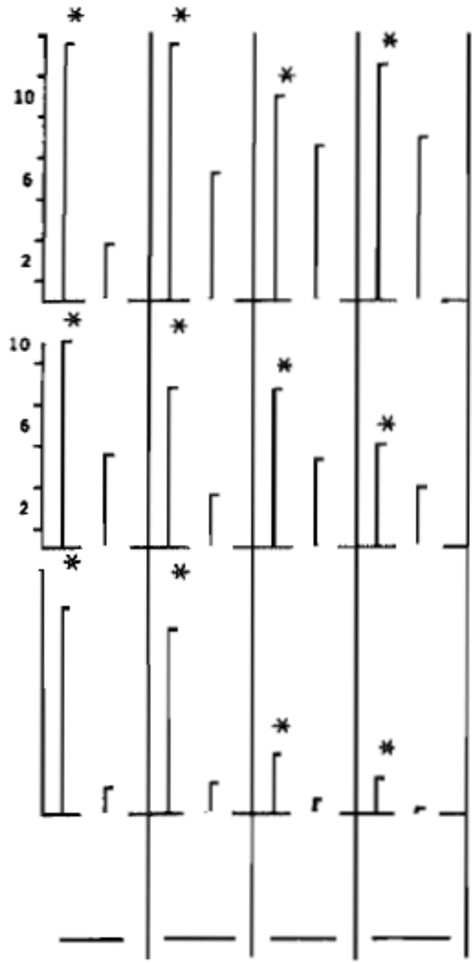
= 1 – proportion of a graphic that can be erased without loss of data-information

(Radical) Tufte Example

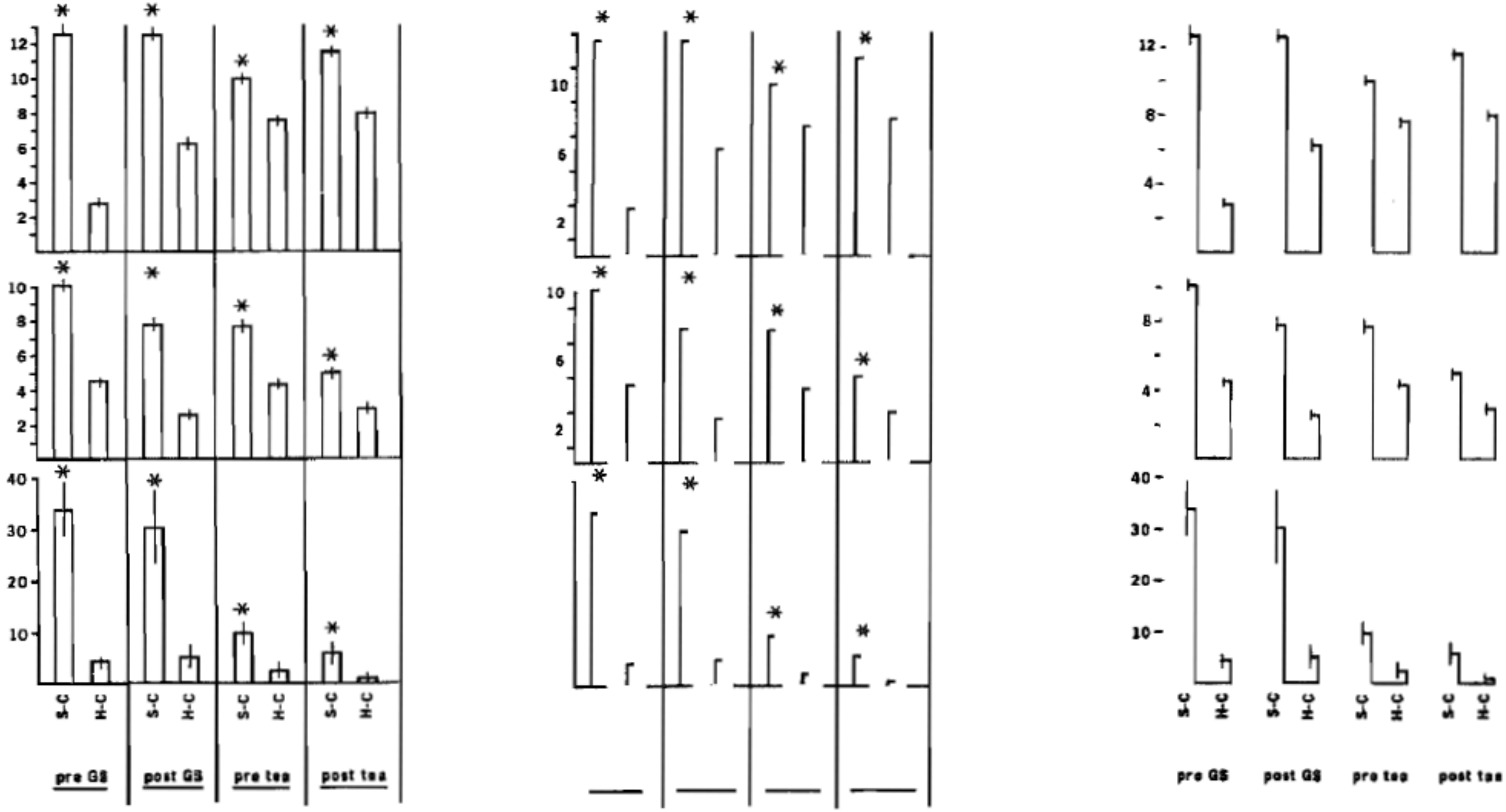


(Radical) Tufte Example





(Radical) Tufte Example



Stephen Few's Design Recommendation

Reduce the Non-Data Pixels

1. Subtract unnecessary non-data pixels.

   Ask yourself: "Would the data suffer any loss of meaning or impact if this were eliminated?"

   If the answer is "no," then get rid of it.

2. De-emphasize and regularize the remaining non-data pixels.

   e.g. use thin lines and light grey for supporting non-data components of the graph (axes, labels, etc.)

Stephen Few's Design Recommendation

Enhance the Data Pixels

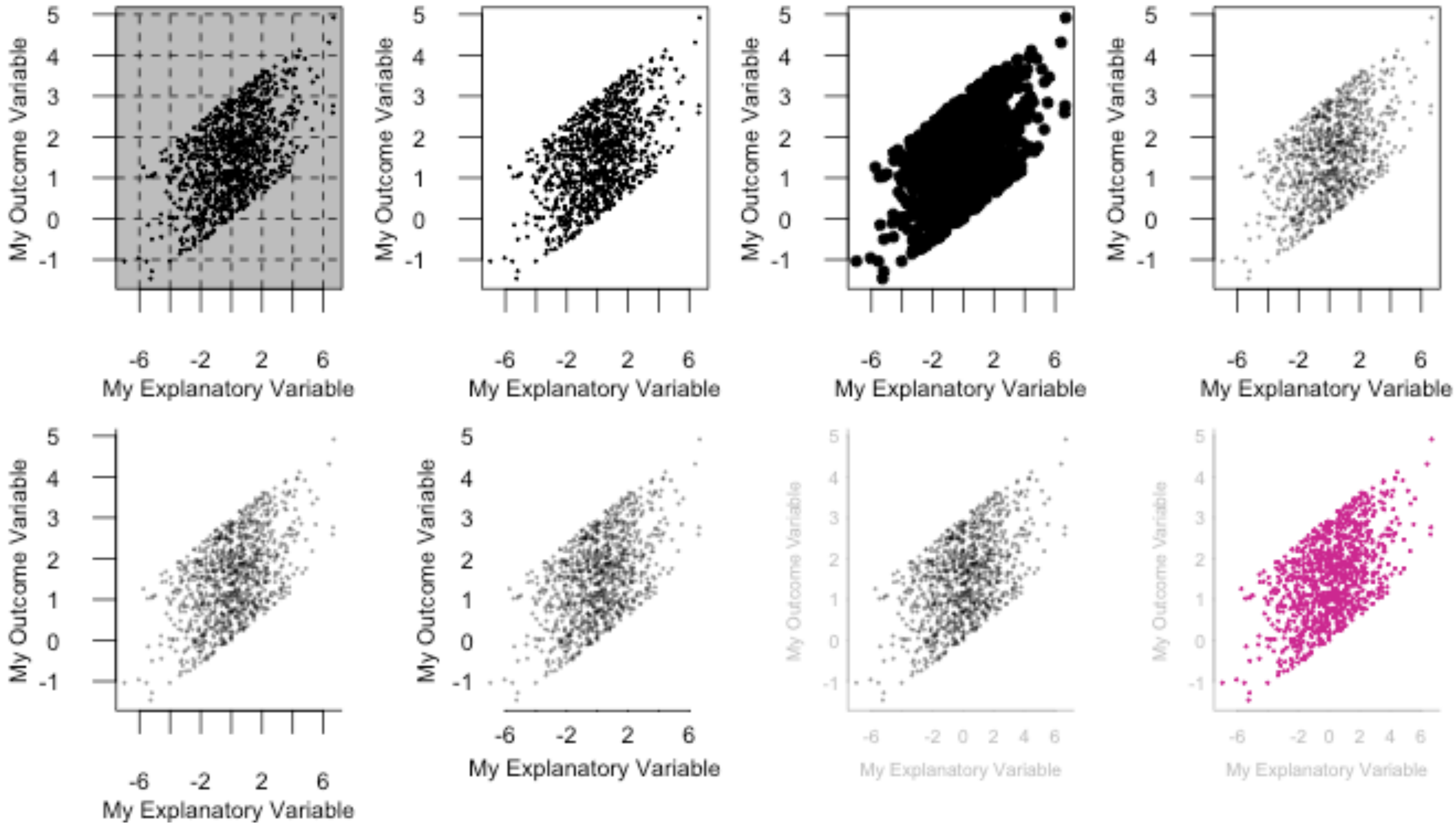
1. Subtract unnecessary data pixels

   Not all information is equally important.

2. Emphasize the most important data pixels

   Use attributes of pre-attentive processing (e.g. color, size, width) to emphasize the most important data pixels.

Example



My Outcome Variable
My Explanatory Variable
5
4
3
2
1
0
-1
-6
-2
2
6
0
4

Comparison, Comparison, Comparison

Comparison, Comparison, Comparison

“The fundamental analytical act in statistical reasoning is to answer the question ´compared to what?´

Whether we are evaluating changes over space or time, searching big data bases, adjusting and controlling for variables, designing experiments, specifying multiple regressions, or doing just about any kind of evidence-based reasoning, the essential point is to make intelligent and appropriate comparisons.

Thus visual displays, if they are to assist thinking, should show comparisons.”
(Tufte 2006: 127)

Some Visual Comparisons

Comparing before and after

Comparing to a standard

Comparing to context

Comparing by subgroups

Comparing to an implicit model ("what do we expect to see?")

Some Graphical Principles for Comparisons

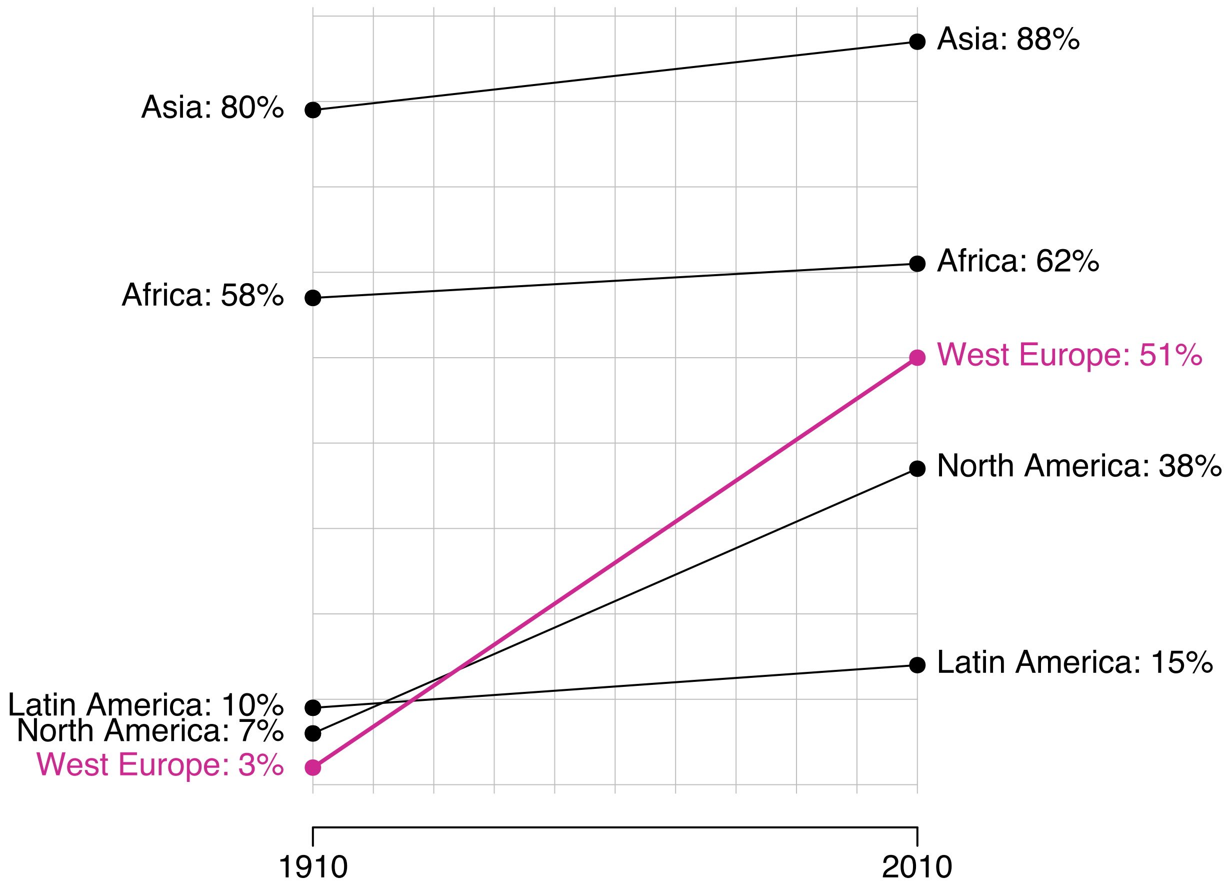
Should the comparison be made within a single graph or across multiple graphs?

If multiple graphs, make sure to use:

- Common graph size
- Common scales
- Helpful alignment

Colour and shape can be used to compare groups within and across graphs.

Comparing before and after: Slope Graph

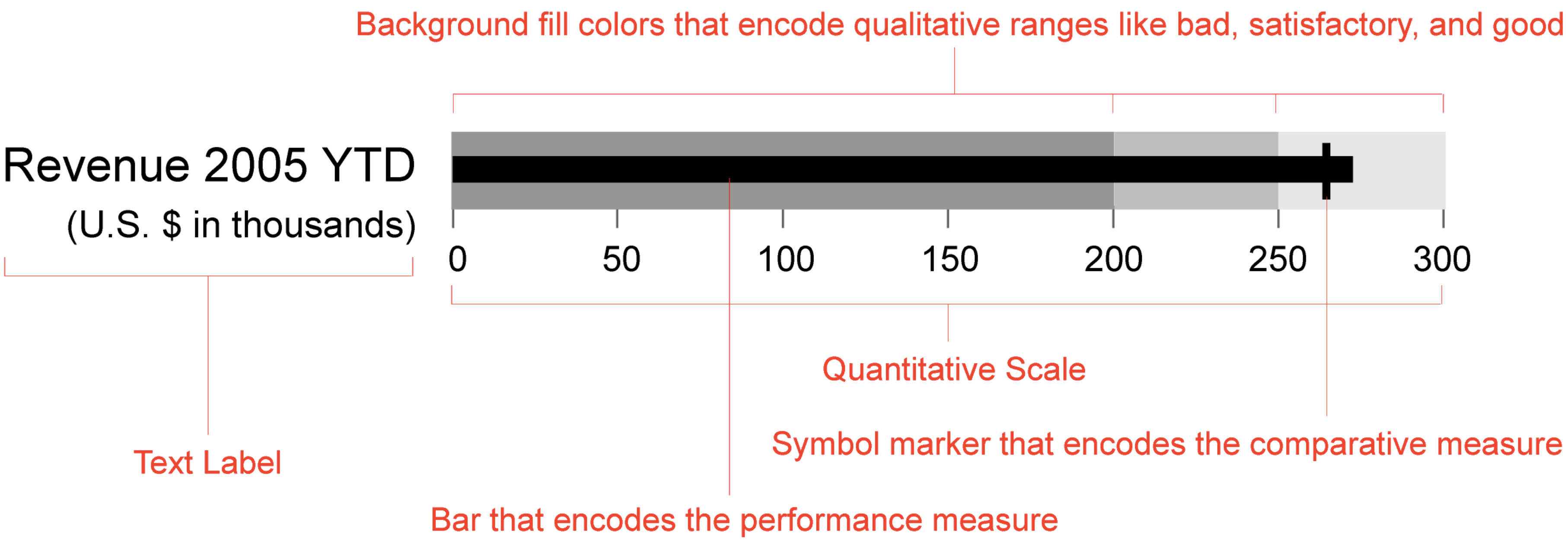


Comparing to a Standard: Bullet Graph

Bullet graphs compare a quantitative measure to

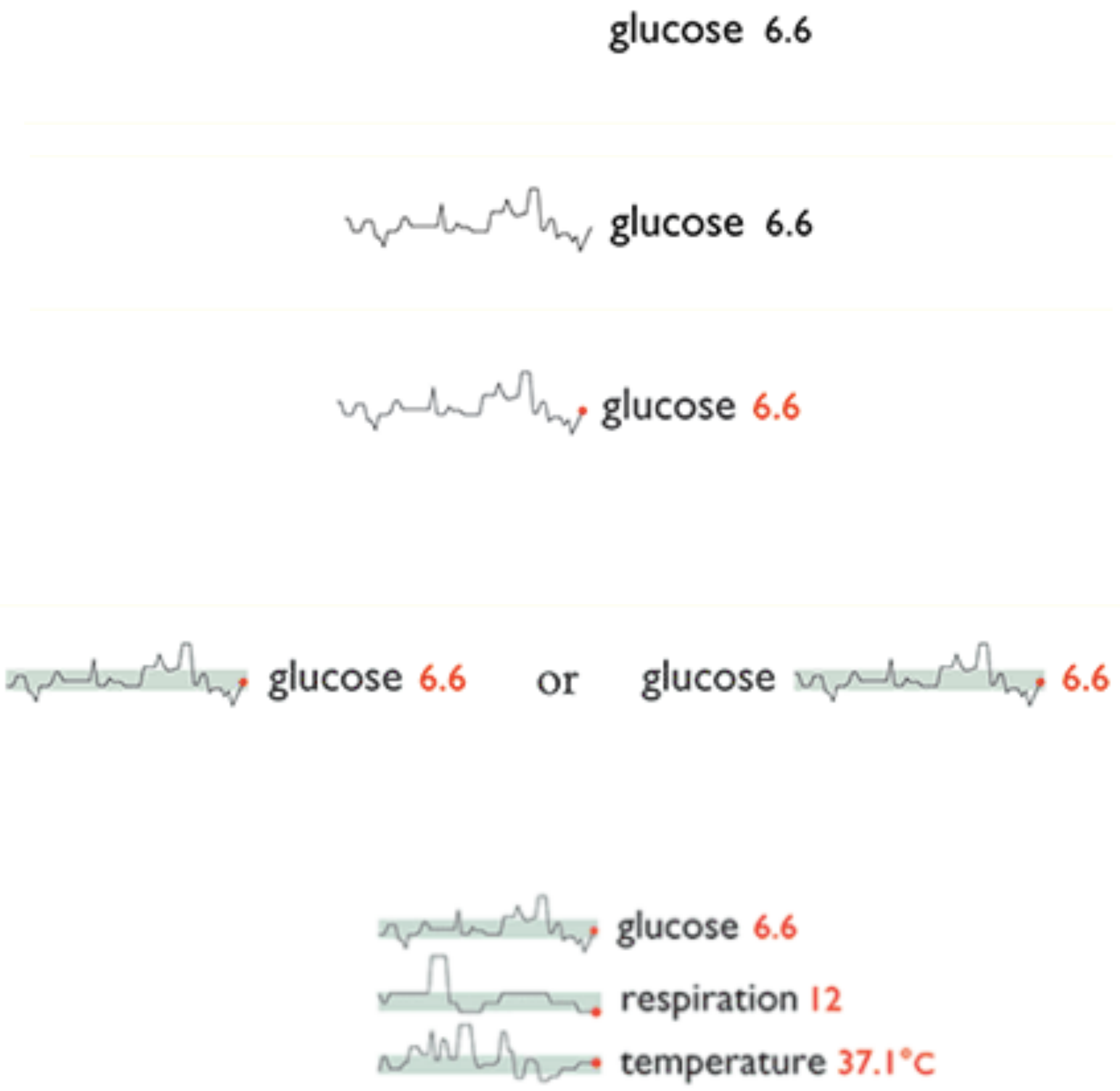
a) one or more related measures (e.g., a target or the same measure in the past) and

b) relate the measure to defined ranges that declare its qualitative state (e.g. good, satisfactory, and poor).



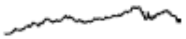
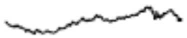
Comparing to a Standard: Bullet Graph

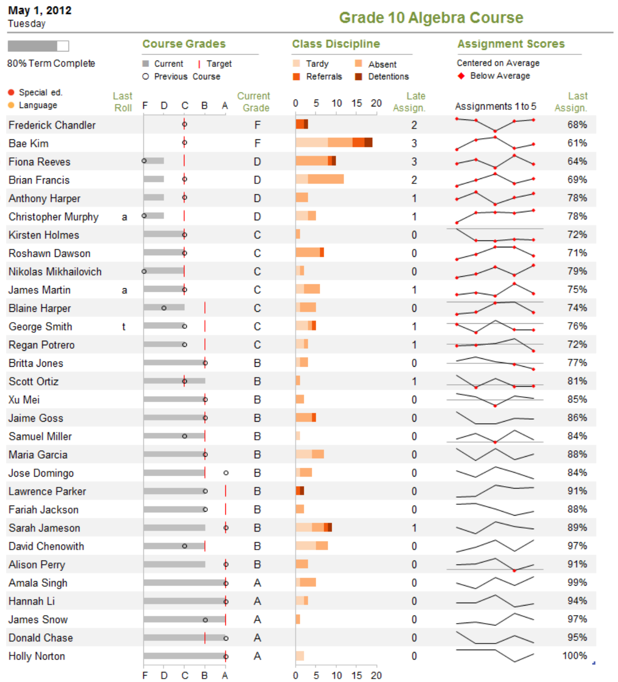
Comparing to Context: Spark Lines



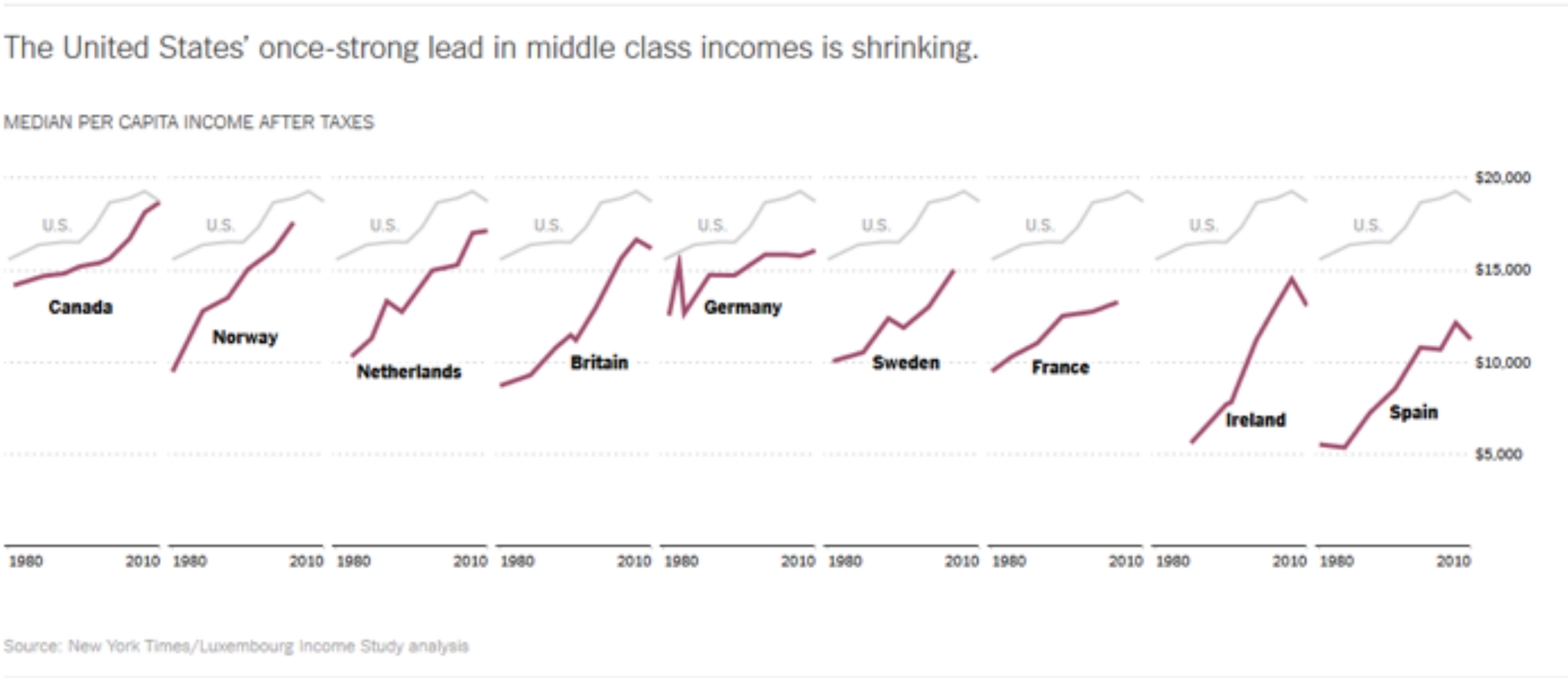
In text paragraph

In table

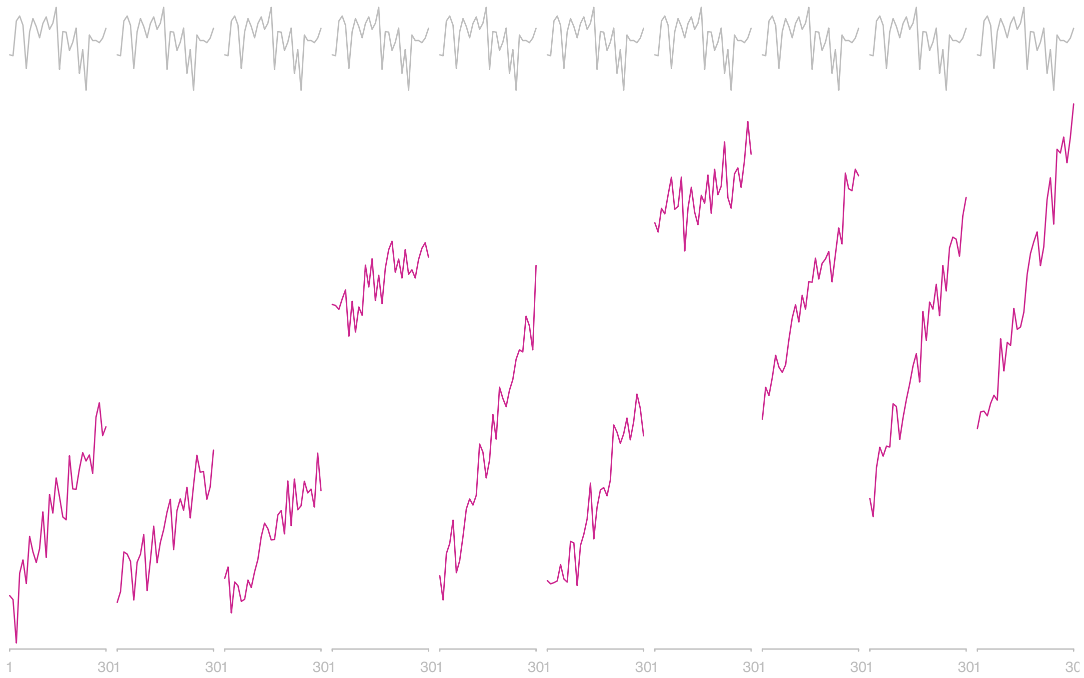
Iris, Tante Lisbeths Nichte, blickte von ihrer Zeitung auf. Sie hat ein gutes Zahlengedächtnis. „Als der Euro eingeführt wurde, sind wir bei knapp 90 Cent  gestartet“, sagte sie. Stimmt. Zur Einführung des Euro als Bargeld am 1.1.2002 musste man 0,88 Dollar für einen Euro bezahlen. Und nochmals drei Jahre früher, zur Einführung des Euro als Buchgeld am 01.01.1999, stand der Euro bei 1,17. Seitdem sieht die Entwicklung so aus: 1,17  1,23.
Tante Lisbeth staunte. Wenn der Euro immer mal wieder auf dem Stand von heute ankommt, mal darüber, mal darunter liegt, woher kommt dann die Hysterie? Geholfen bei unserer Betrachtung hat uns Datendichte: Statt



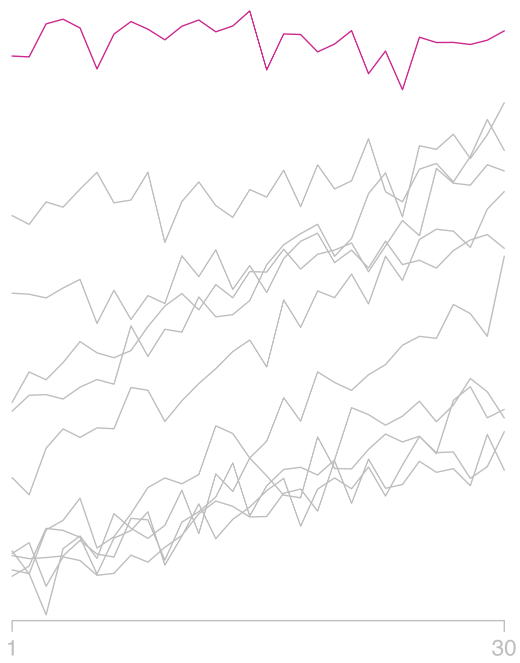
Comparing by Subgroups: Small Multiples



Small Multiples



Single Line Chart



2000: State-level support (orange) or opposition (green) on school vouchers, relative to the national average of 45% support

Orange and green colors correspond to states where support for vouchers was greater or less than the national average.
The seven ethnicreligious cagetories are mutually exclusive. "Evangelicals" includes Mormons as well as born-again Protestants.
Where a category represents less than 1% of the voters of a state, the state is left blank.

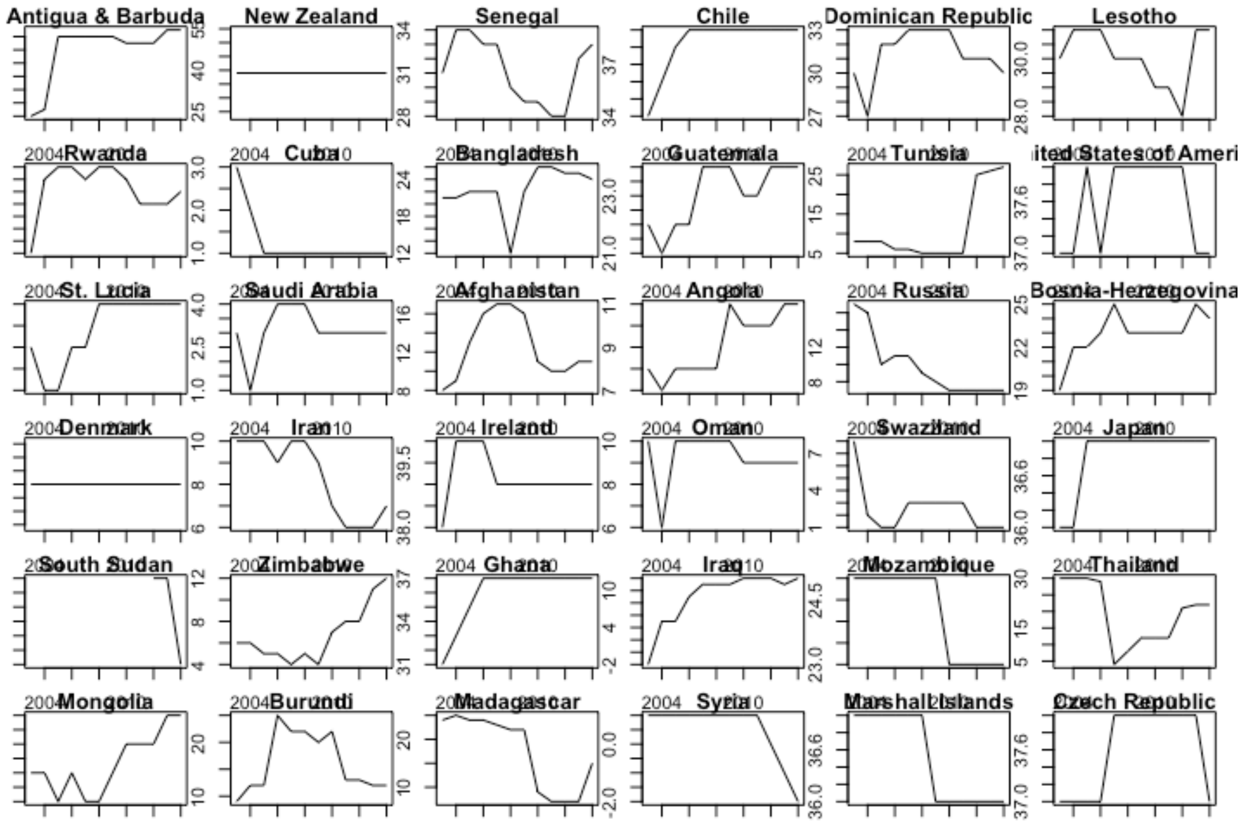
Some Rules for Small Multiple Designs

→ Constant axis scales!

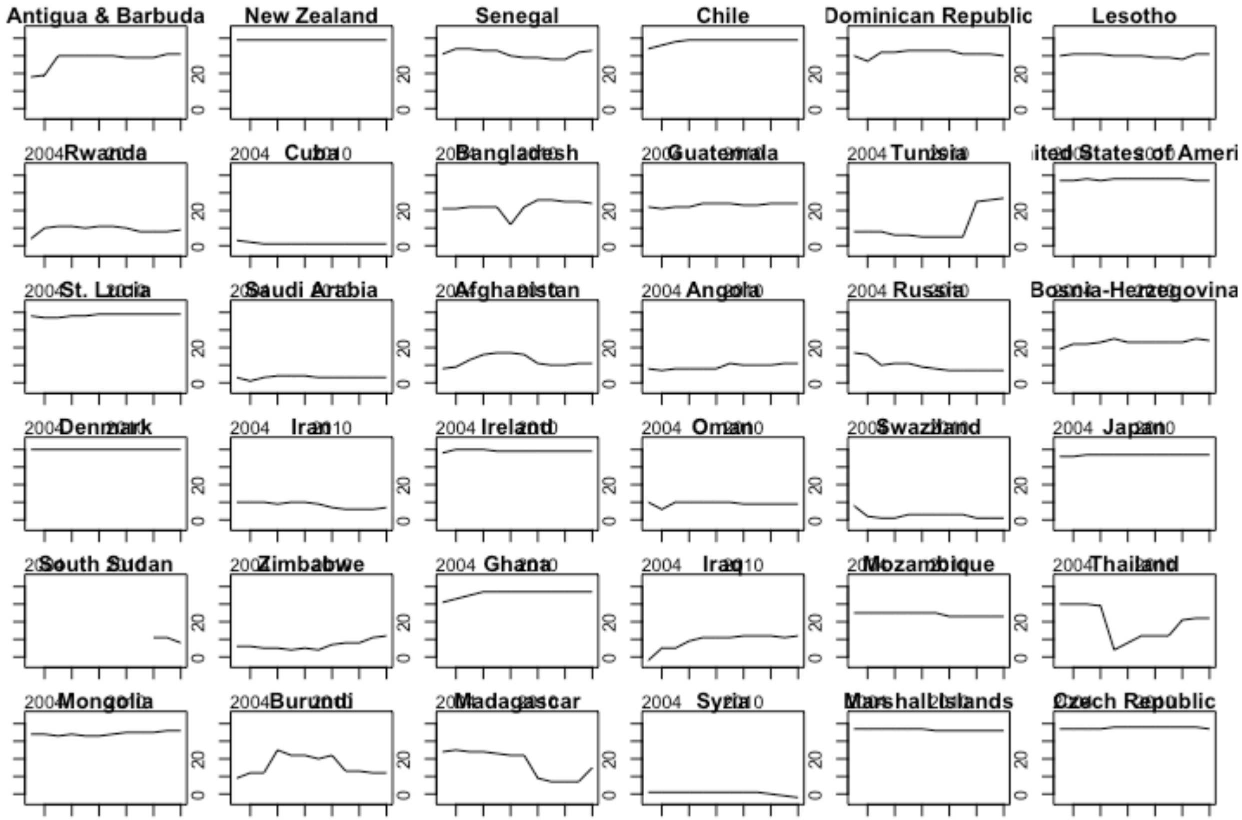
→ Increase data-ink ratio!

→ Sensible Ordering!

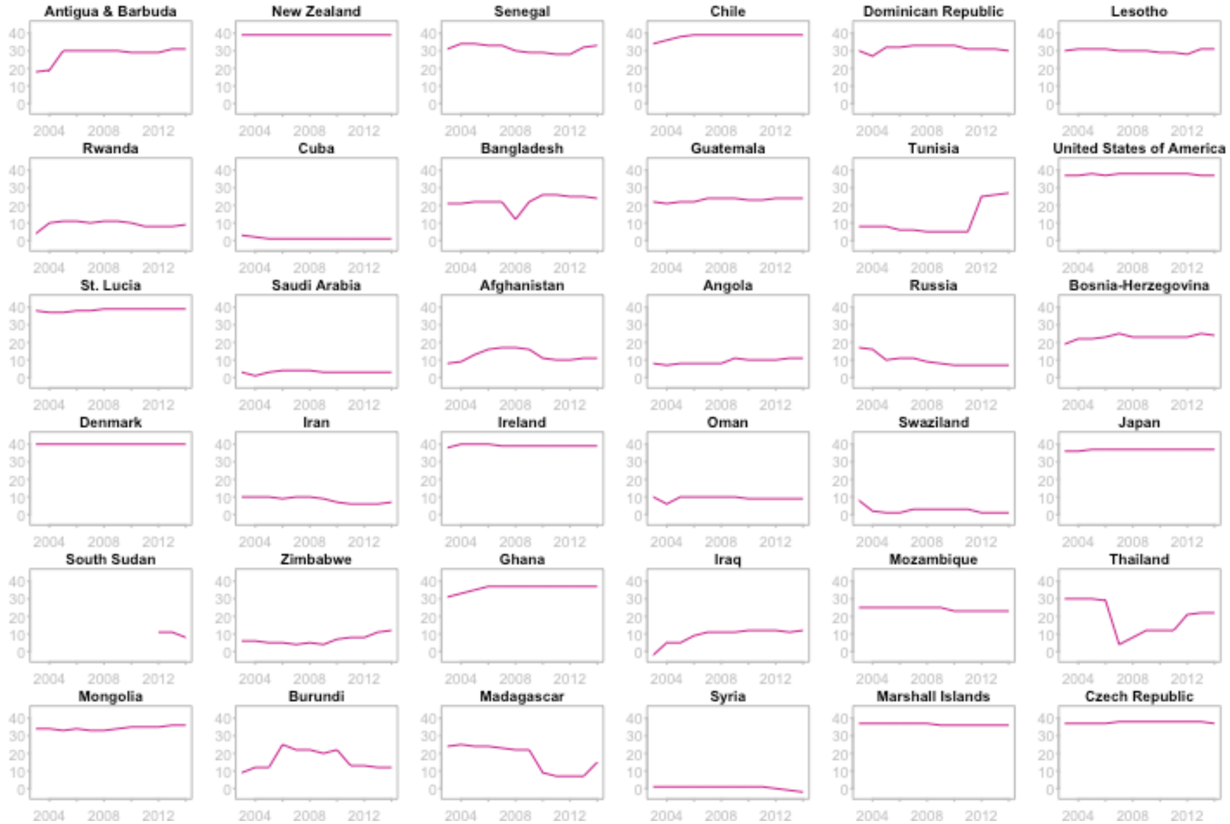
Raw small multiple design



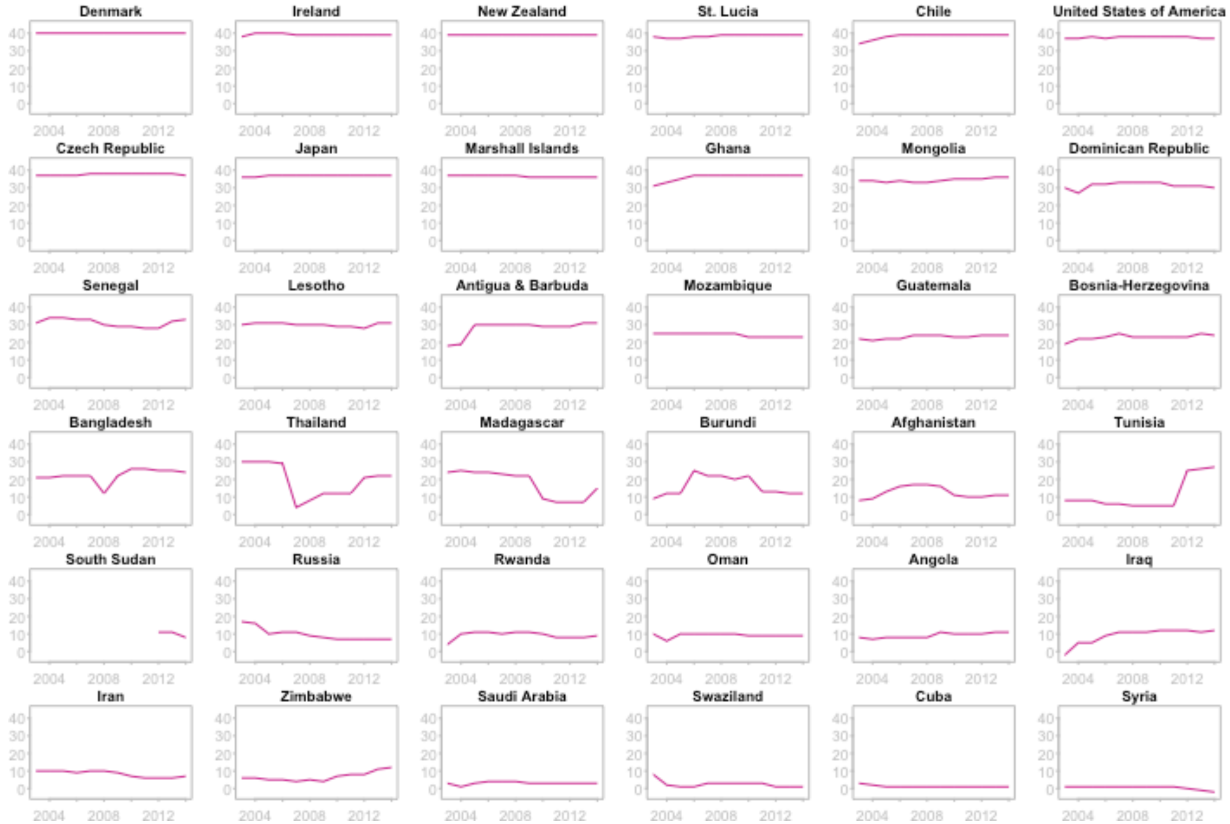
Constant axis scale!



Increase data-ink ratio!



Sensible ordering!

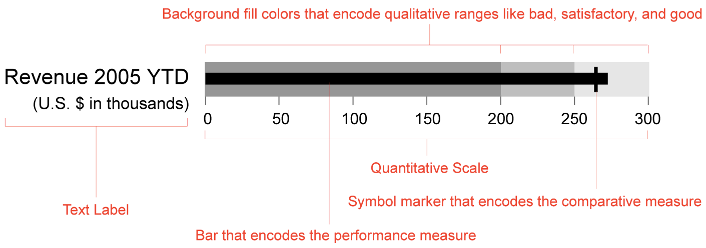


How to do it in R

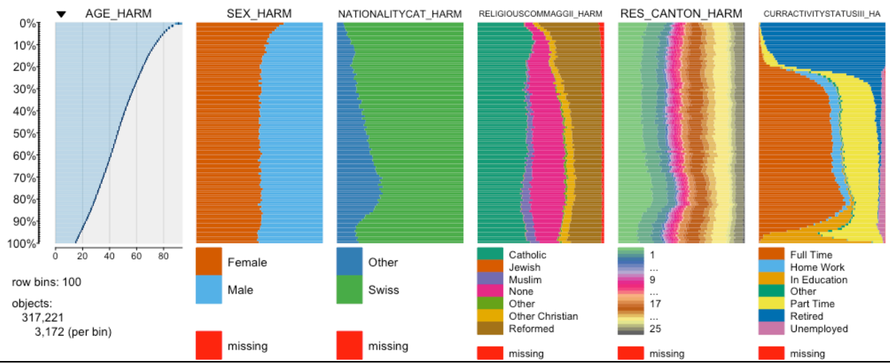
In the Lab

Basic Visualization in R

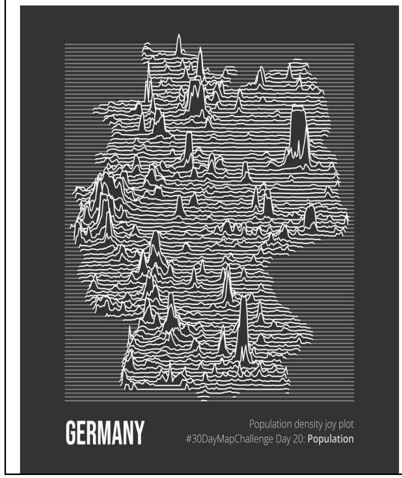
Constructing a Bullet Graph in R



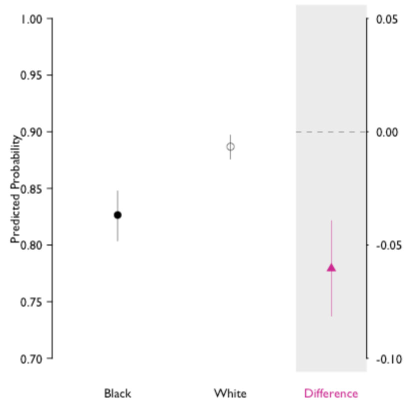
Exploring Data using Table Plots in R



Constructing a Joy Plot Map in R



Visualizing Statistical Models in R



Base Graphics

High-level commands: create a plot of a particular format

Low-level commands: add elements to a plot (points, lines, text, etc.)

Arguments and parameters: define and fine-tune various elements in a plot (color, size, etc.)

```
parameters(arguments)

highlevelcommand(data, arguments)

lowlevelcommand(data, arguments)

lowlevelcommand(data, arguments)

lowlevelcommand(data, arguments)
```

Before Graphing

Set Working Directory

```
setwd("/YOURDIRECTORY/")
```

Getting Data into R

```
library(foreign)

data <- read.csv("gapminder.csv", sep=";")
```

Before Graphing

Quick Look at Data Set

```
head(data)
```

```
      country continent year lifeExp      pop gdpPercap
1 Afghanistan      Asia 1952  28.801  8425333  779.4453
2 Afghanistan      Asia 1957  30.332  9240934  820.8530
3 Afghanistan      Asia 1962  31.997 10267083  853.1007
4 Afghanistan      Asia 1967  34.020 11537966  836.1971
5 Afghanistan      Asia 1972  36.088 13079460  739.9811
6 Afghanistan      Asia 1977  38.438 14880372  786.1134
```

Before Graphing

Look at Single Variables

```
class(data$lifeExp)

[1] "numeric"


class(data$continent)

[1] "factor"


summary(data$lifeExp)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.60   48.20   60.71   59.47   70.85   82.60


summary(data$continent)

  Africa Americas     Asia   Europe  Oceania
     624      300      396      360       24
```

Before Graphing

Creating a Simple Table

```
table(data$continent)

  Africa Americas     Asia   Europe  Oceania 
     624      300      396      360       24 
```

```
my.tab <- table(data$continent)
prop.table(my.tab)

    Africa   Americas       Asia     Europe    Oceania 
0.36619718 0.17605634 0.23239437 0.21126761 0.01408451 
```

Before Graphing

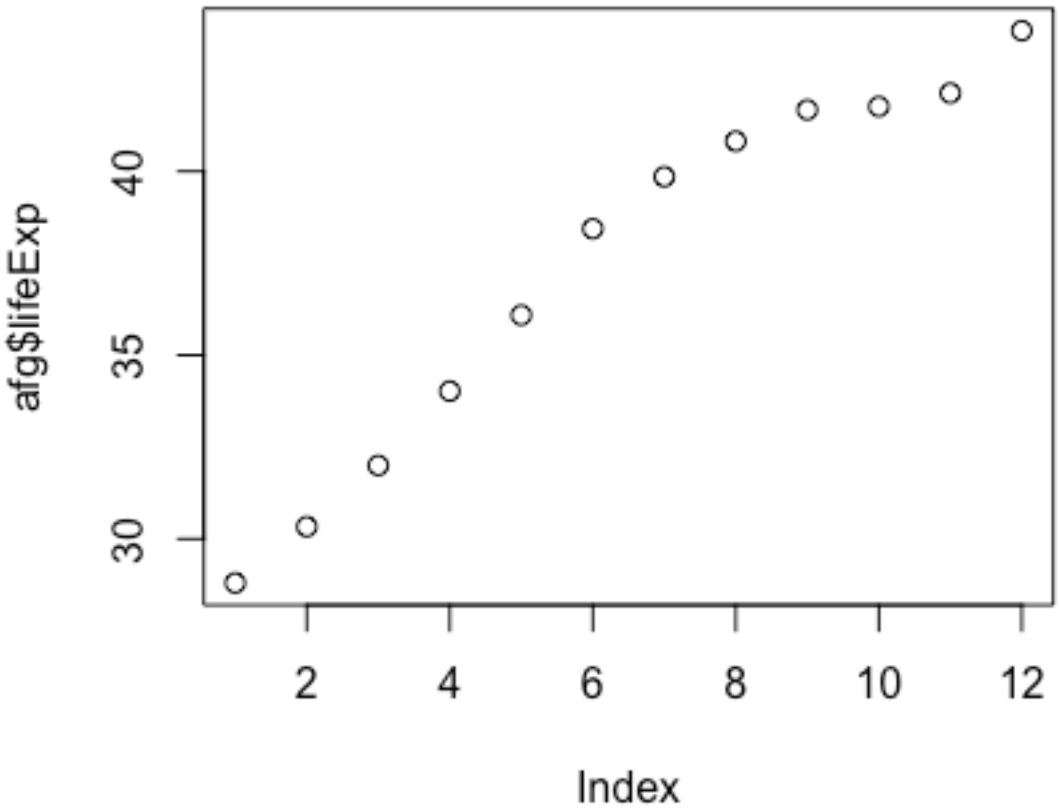
Subsetting the Data Set

```
afg <- data[data$country=="Afghanistan",]
```

Creating Plots Using High-level Functions

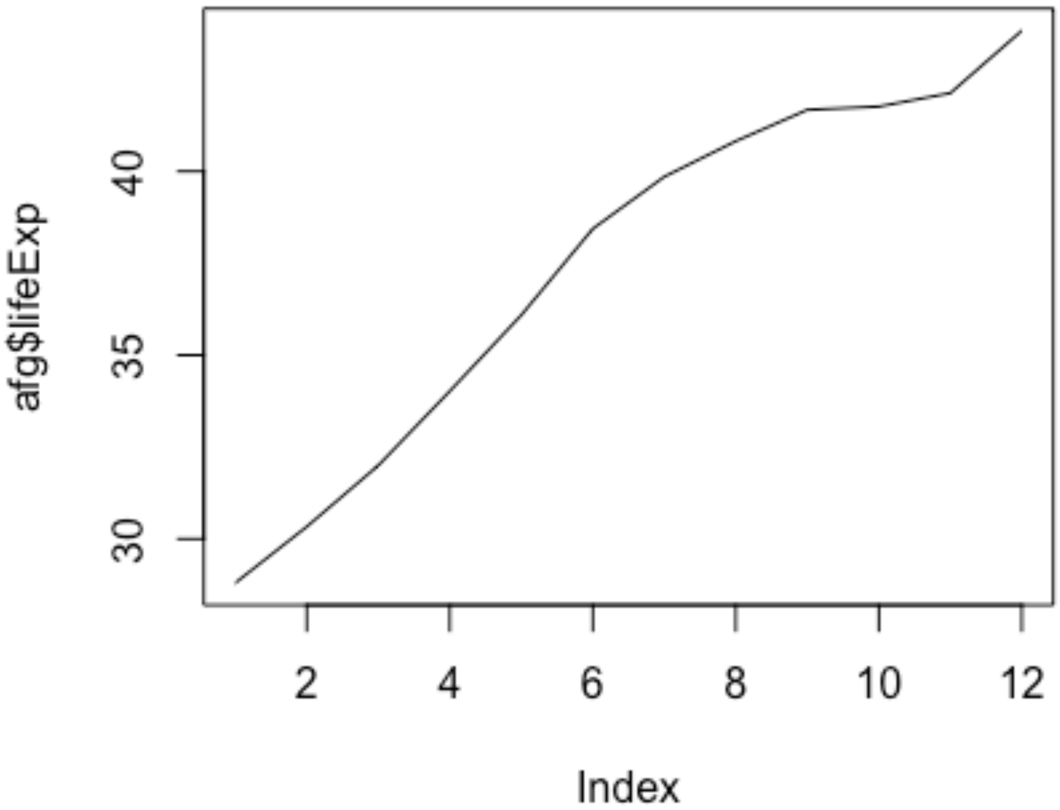
Some High-level Functions

```
plot(afg$lifeExp)
```



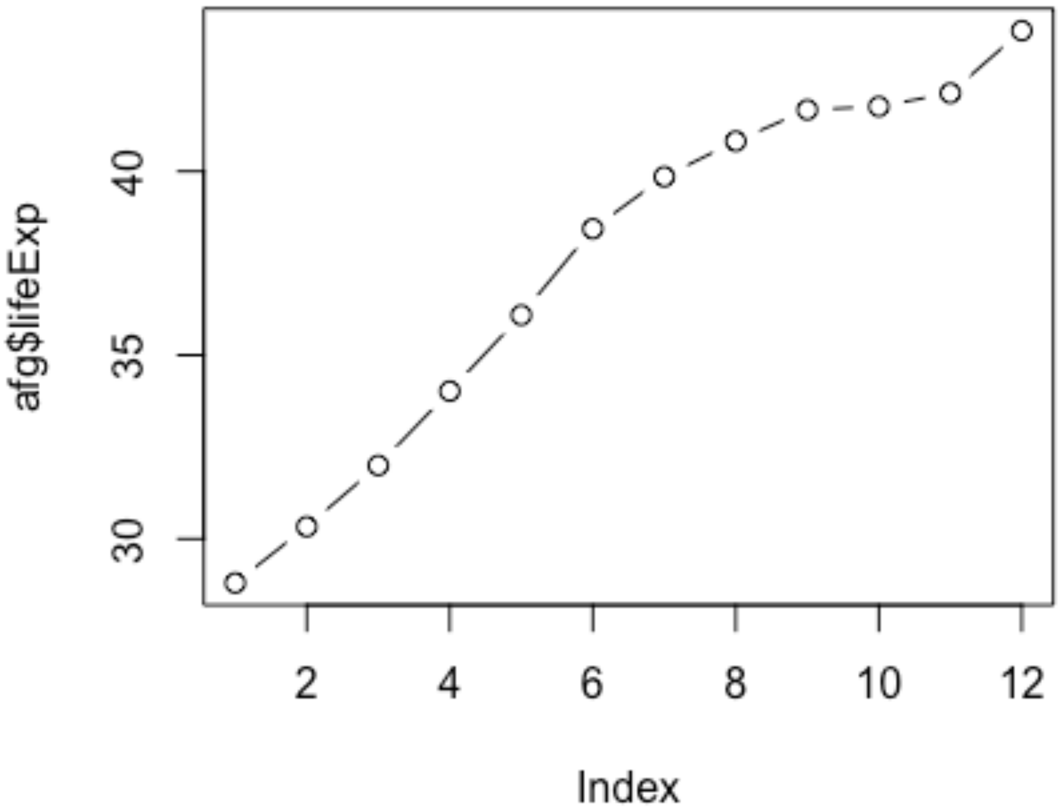
Some High-level Functions

```
plot(afg$lifeExp, type="l")
```



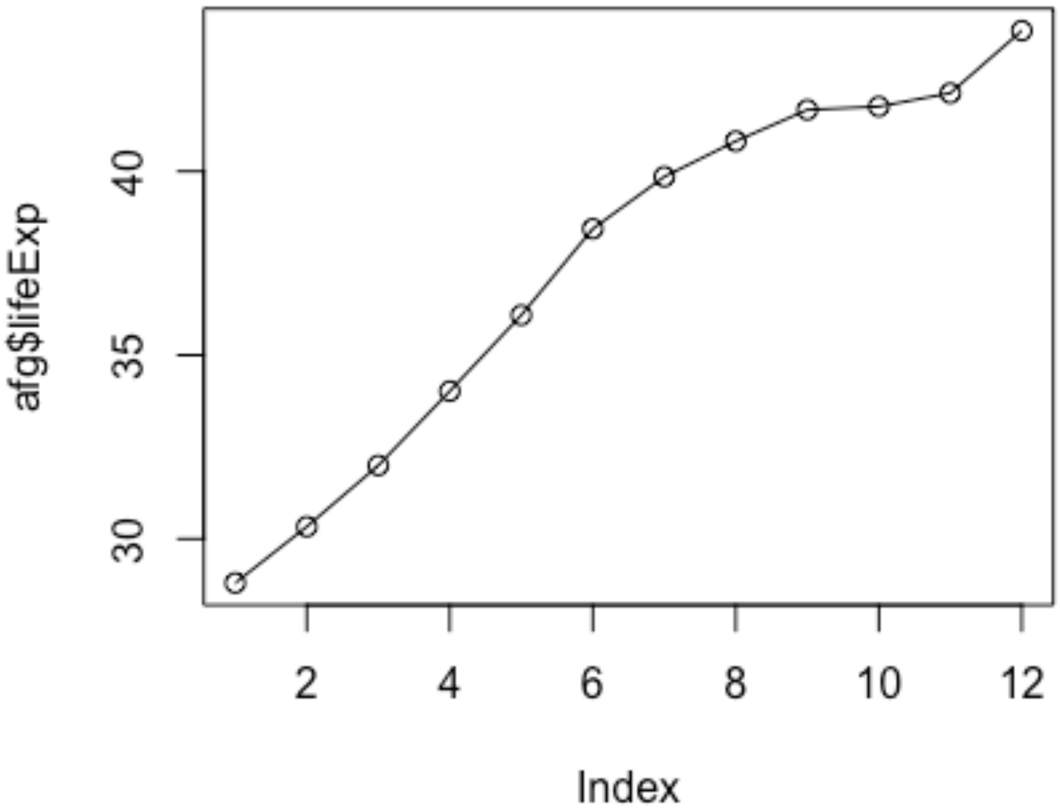
Some High-level Functions

```
plot(afg$lifeExp, type="b")
```



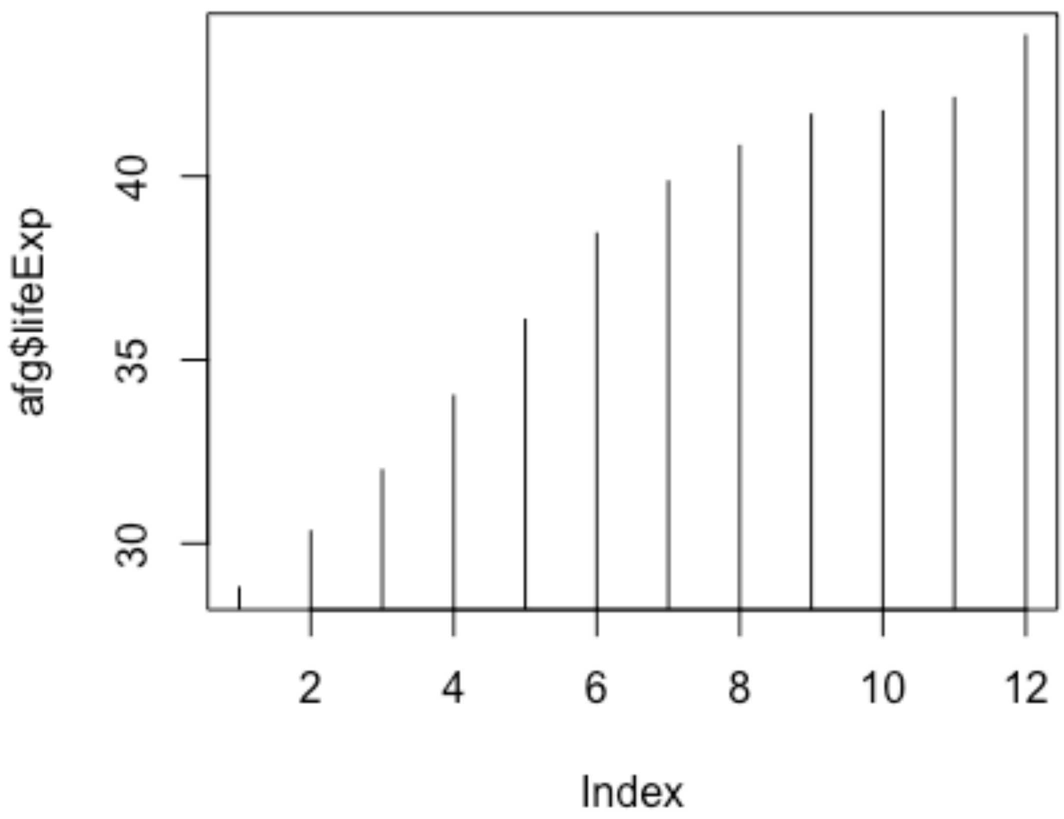
Some High-level Functions

```
plot(afg$lifeExp, type="o")
```



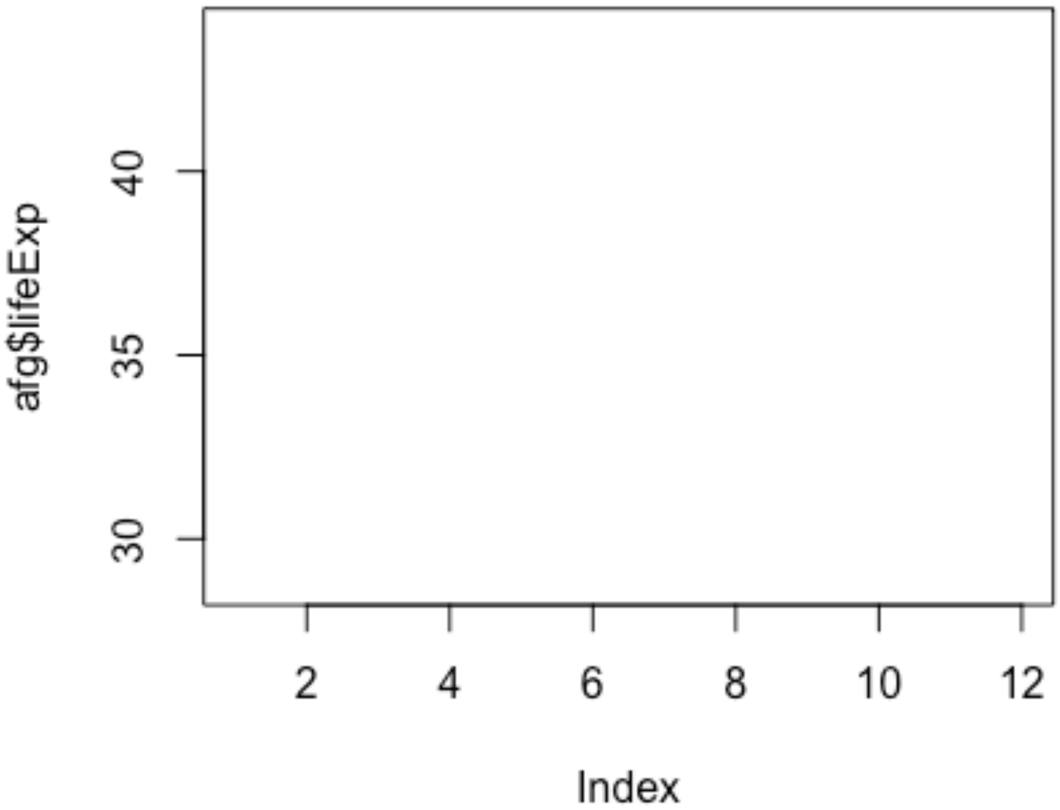
Some High-level Functions

```
plot(afg$lifeExp, type="h")
```



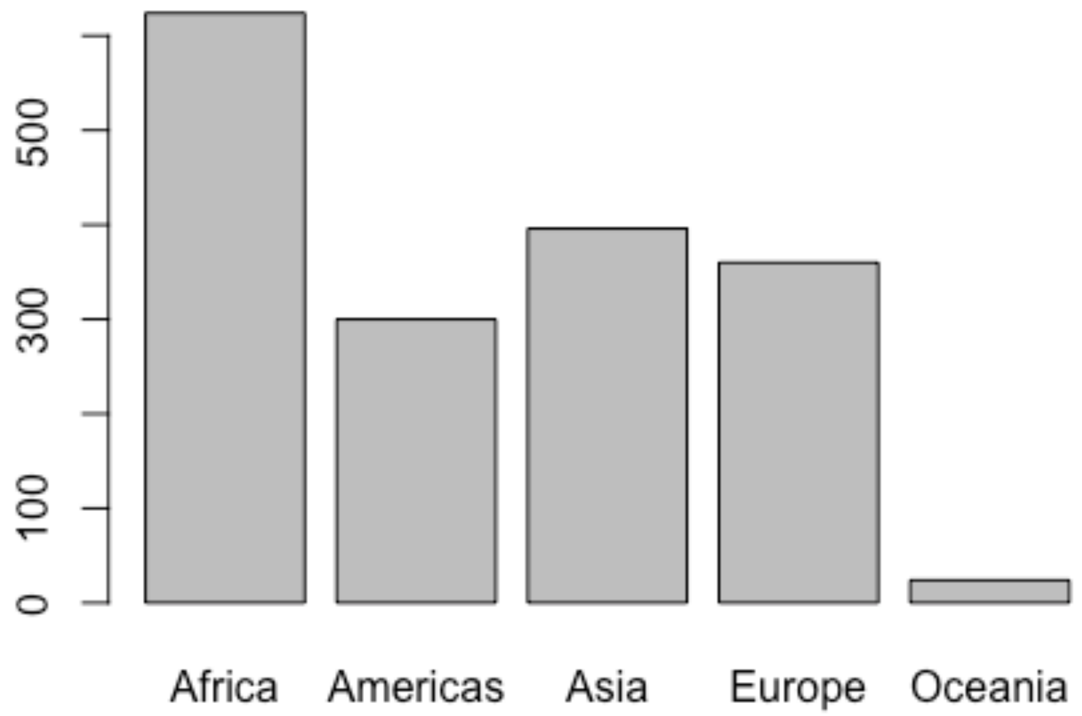
Some High-level Functions

```
plot(afg$lifeExp, type="n")
```



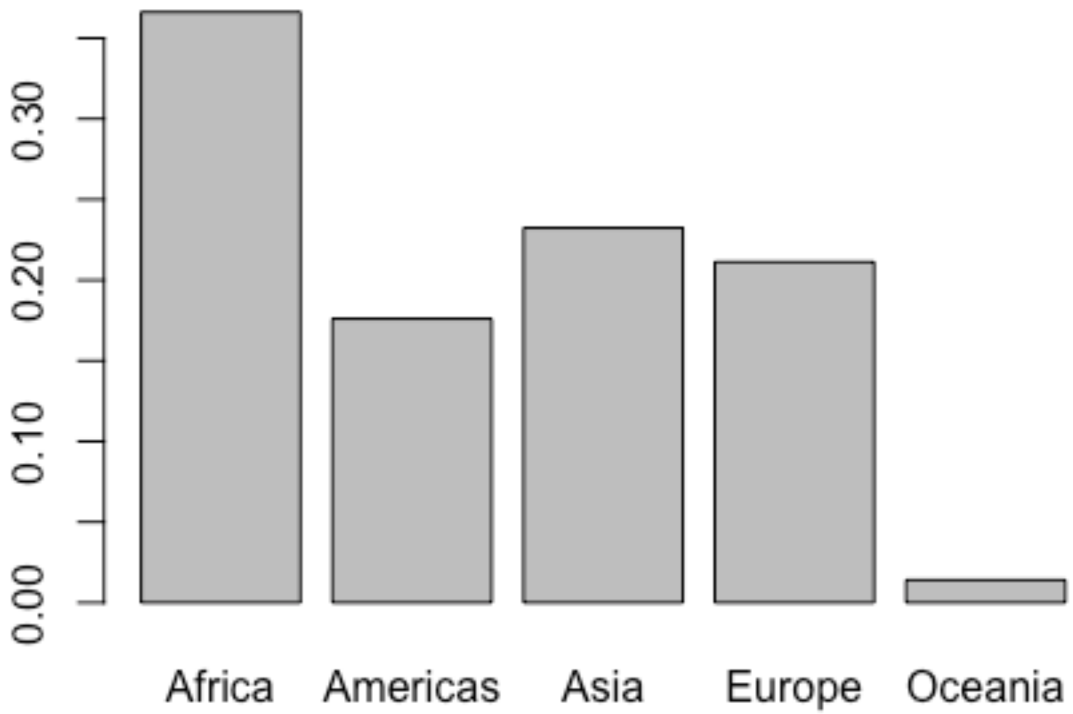
Some High-level Functions

```
plot(data$continent)
```



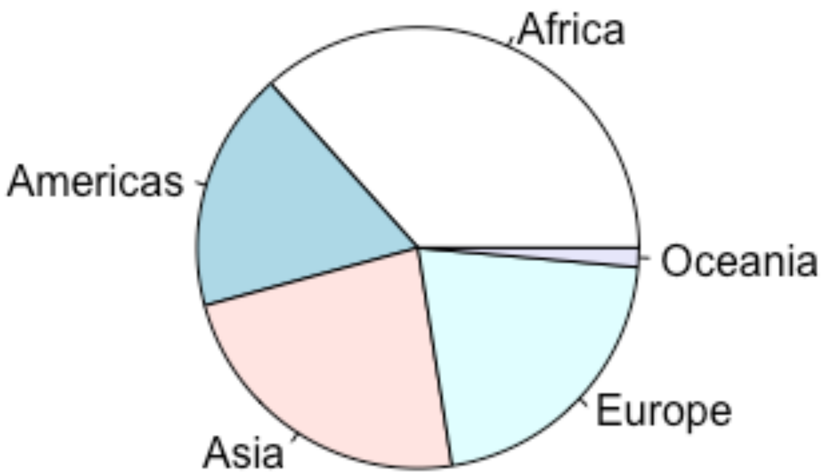
Some High-level Functions

```
my.tab <- prop.table(table(data$continent))
barplot(my.tab)
```



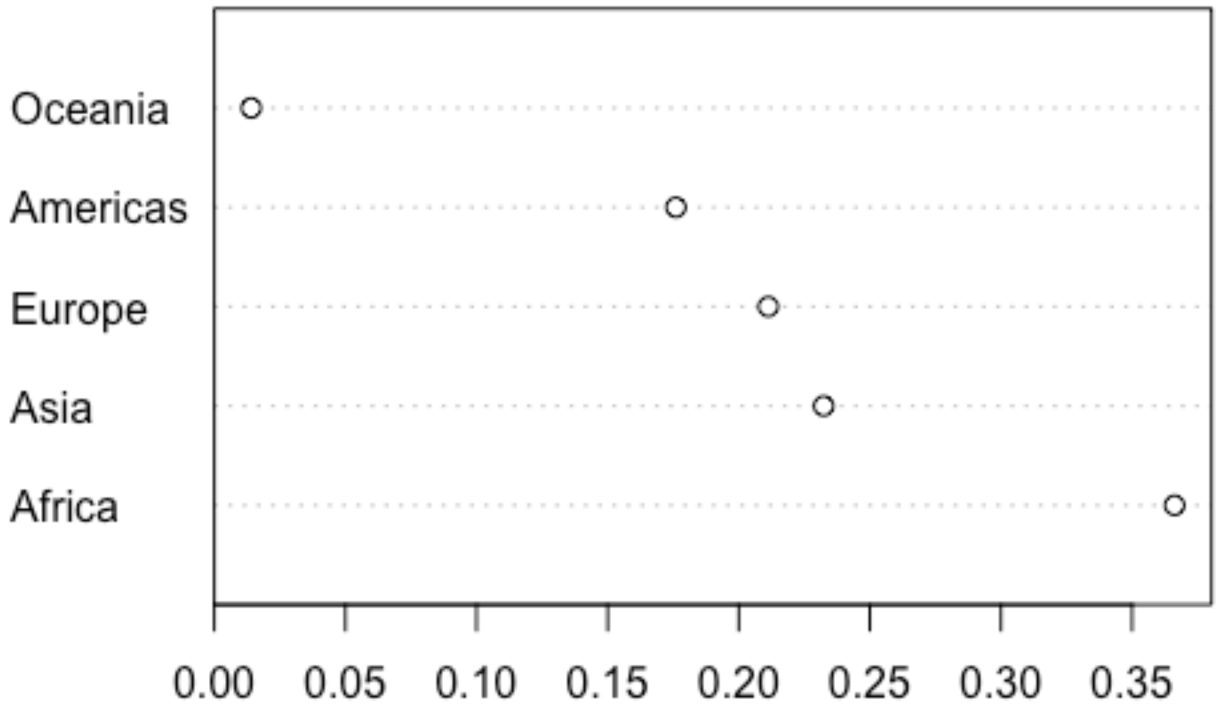
Some High-level Functions

```
pie(my.tab)
```

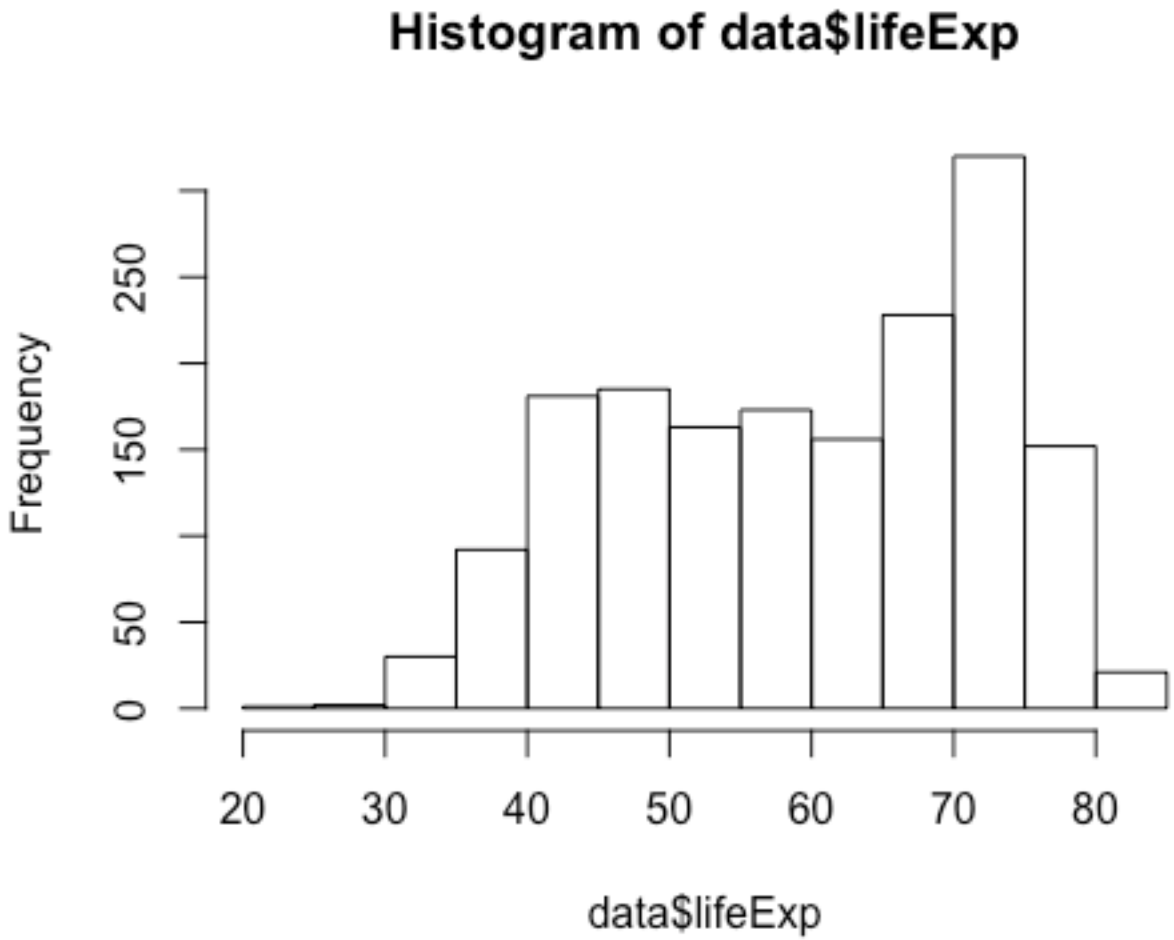


Some High-level Functions

```
dotchart(my.tab)
```



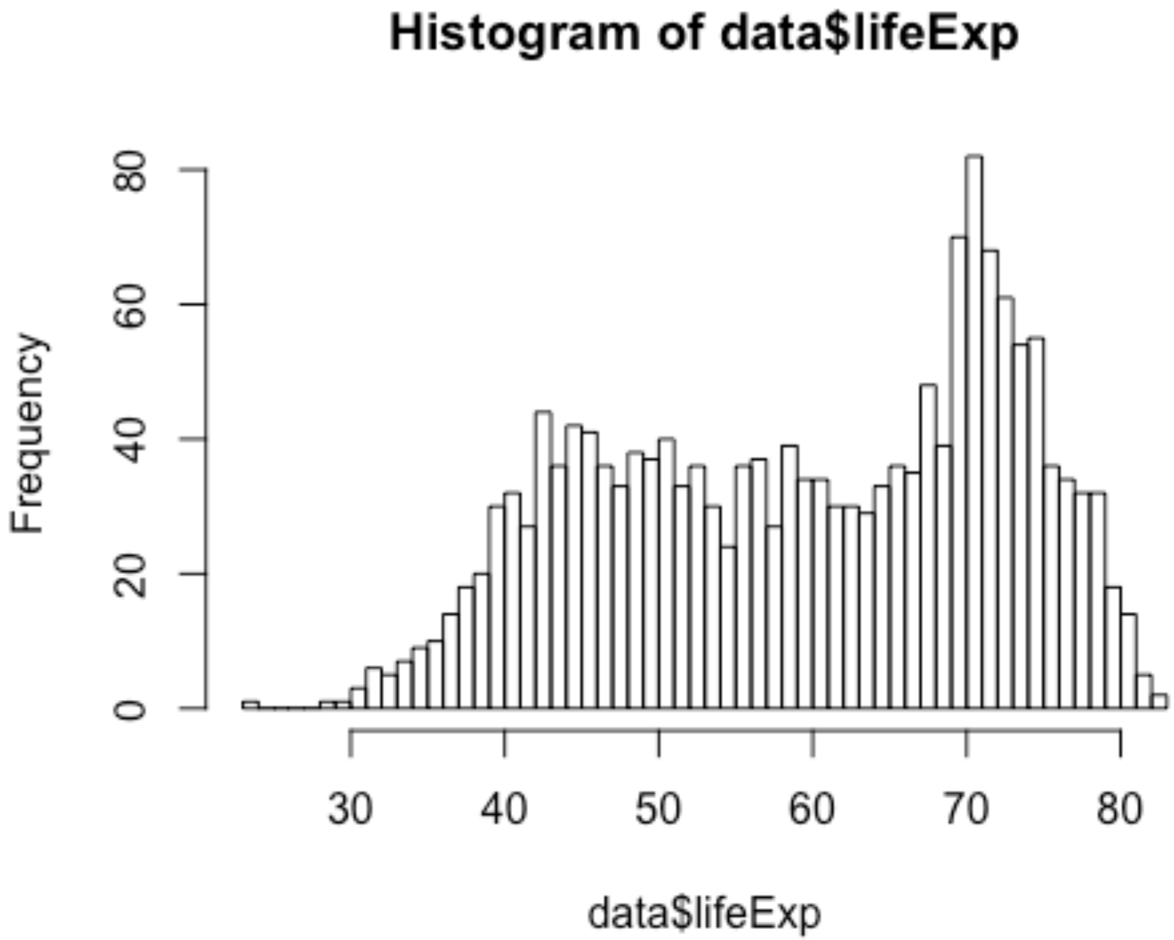
```
hist(data$lifeExp)
```



```
hist(data$lifeExp, breaks=100)
```

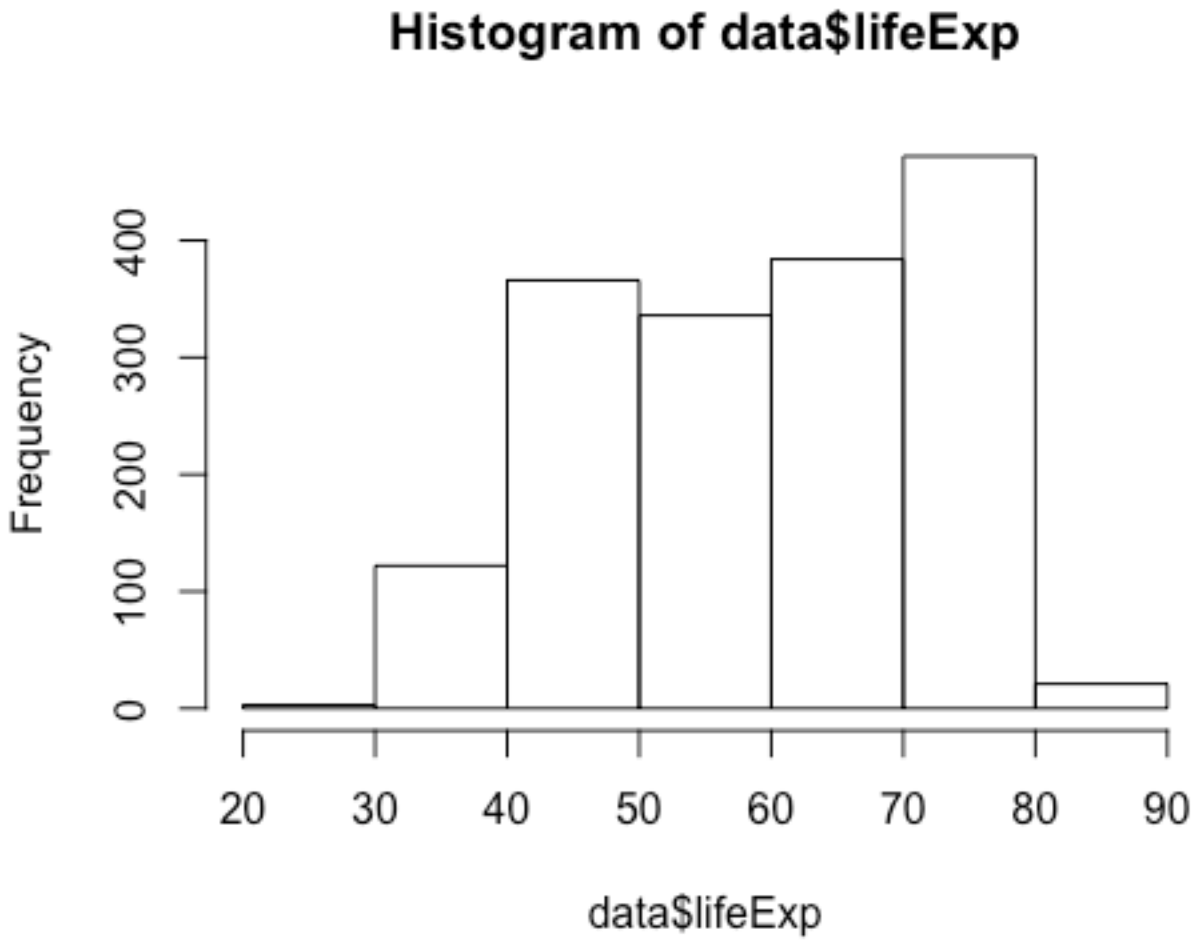
Some High-level Functions

```
hist(data$lifeExp, breaks=50)
```



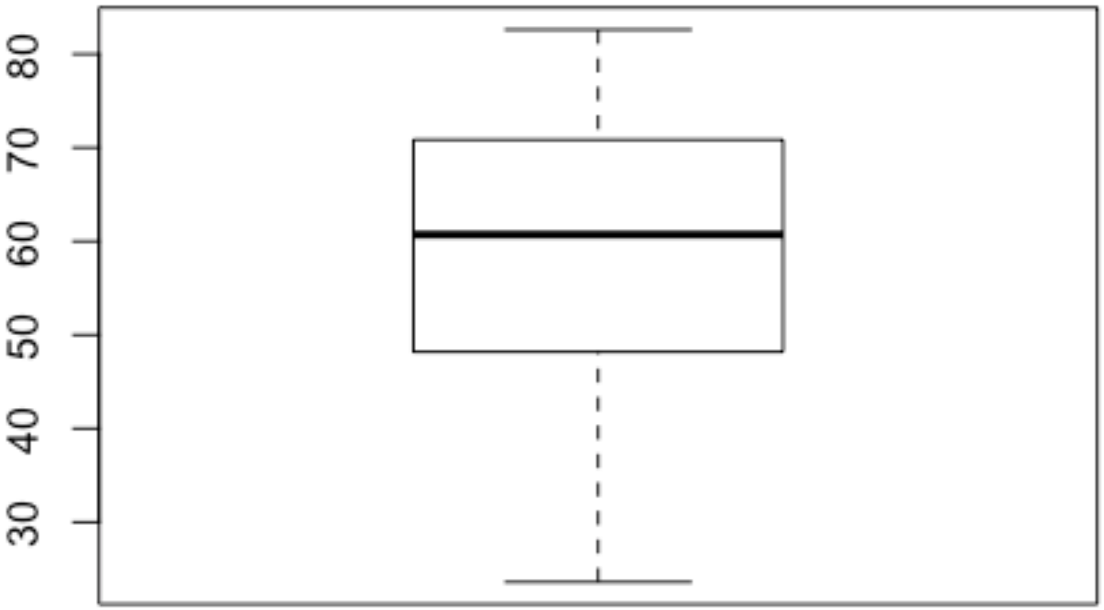
Some High-level Functions

```
hist(data$lifeExp, breaks=5)
```



Some High-level Functions

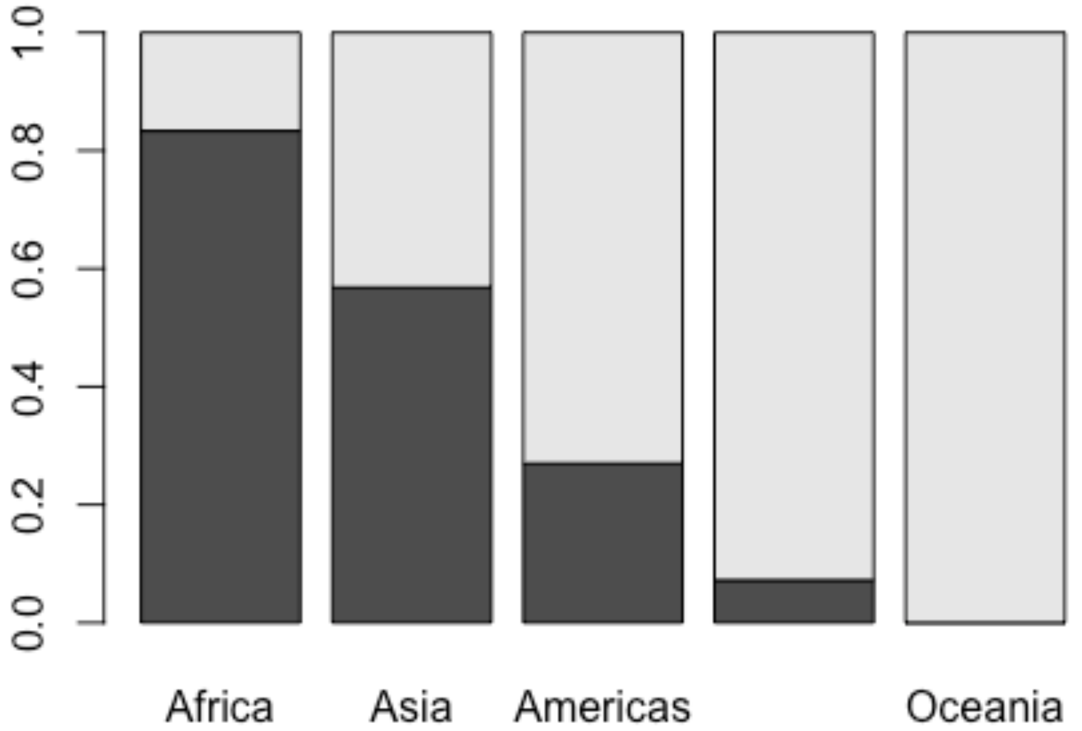
```
boxplot(data$lifeExp)
```



Changing Elements of a Graphic

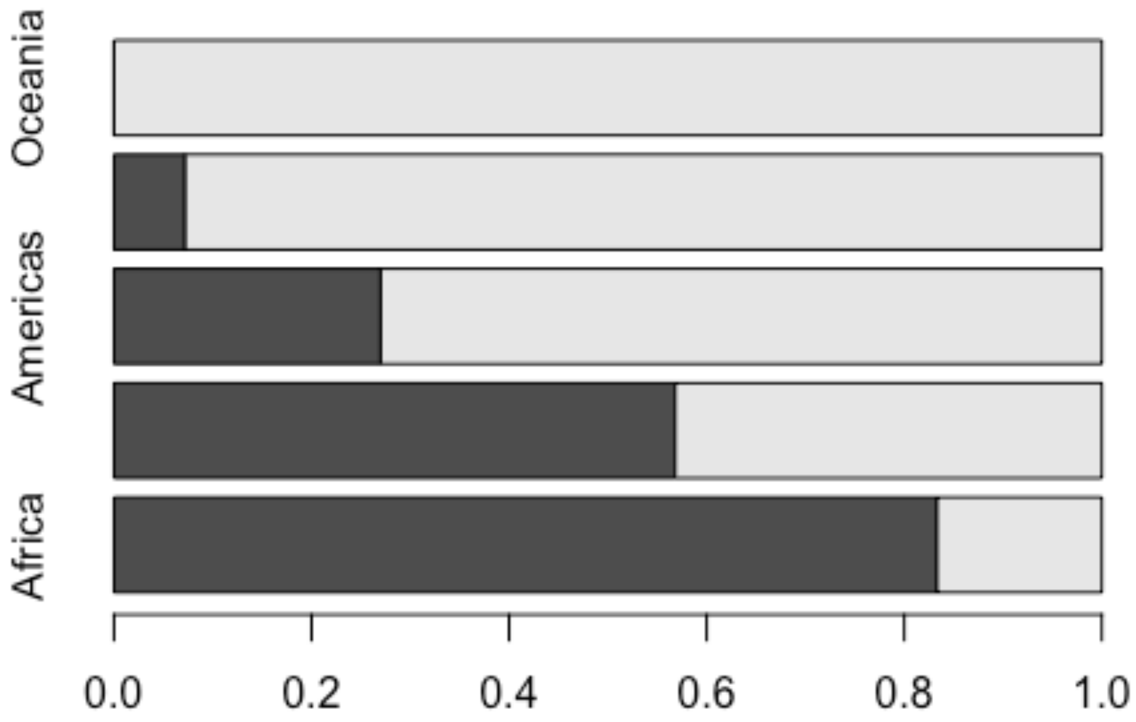
Some Arguments: Graphical Parameters

```
barplot(my.tab)
```



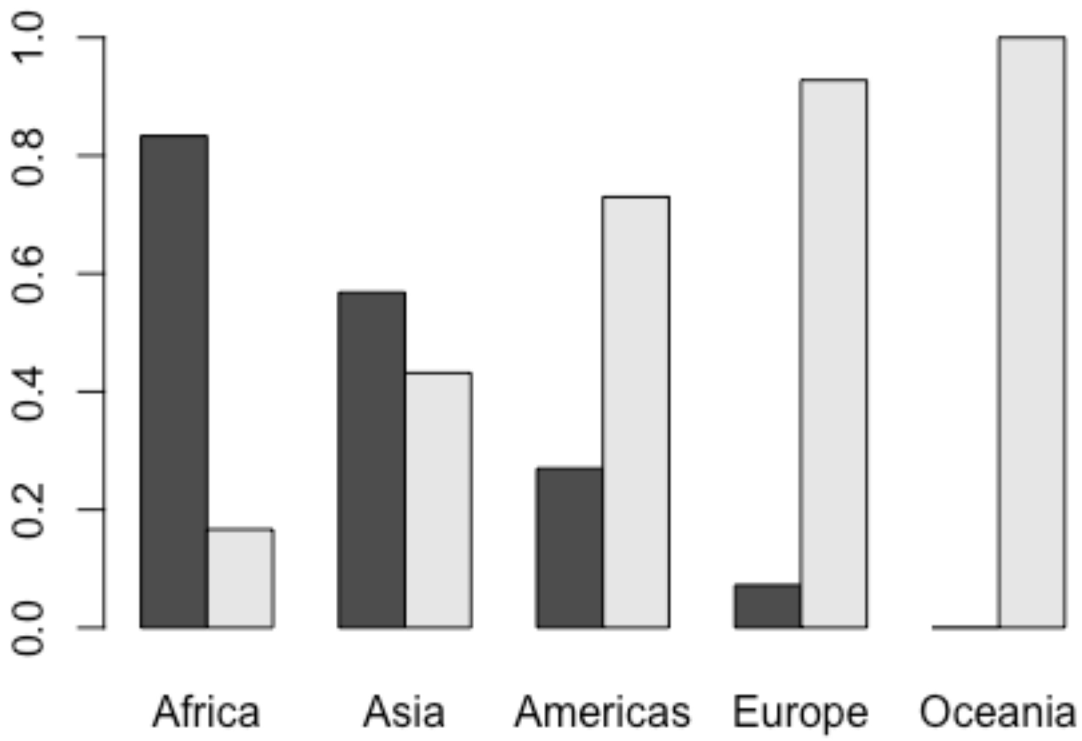
Some Arguments: Graphical Parameters

```
barplot(my.tab, horiz=T)
```



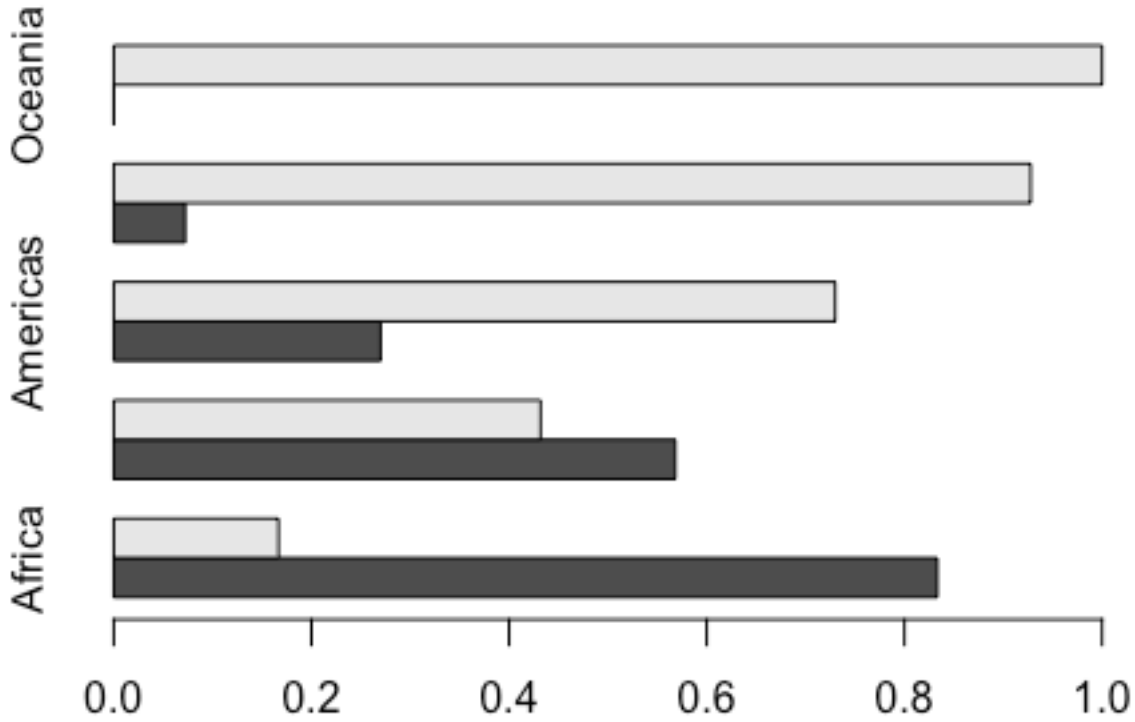
Some Arguments: Graphical Parameters

```
barplot(my.tab, beside=T)
```



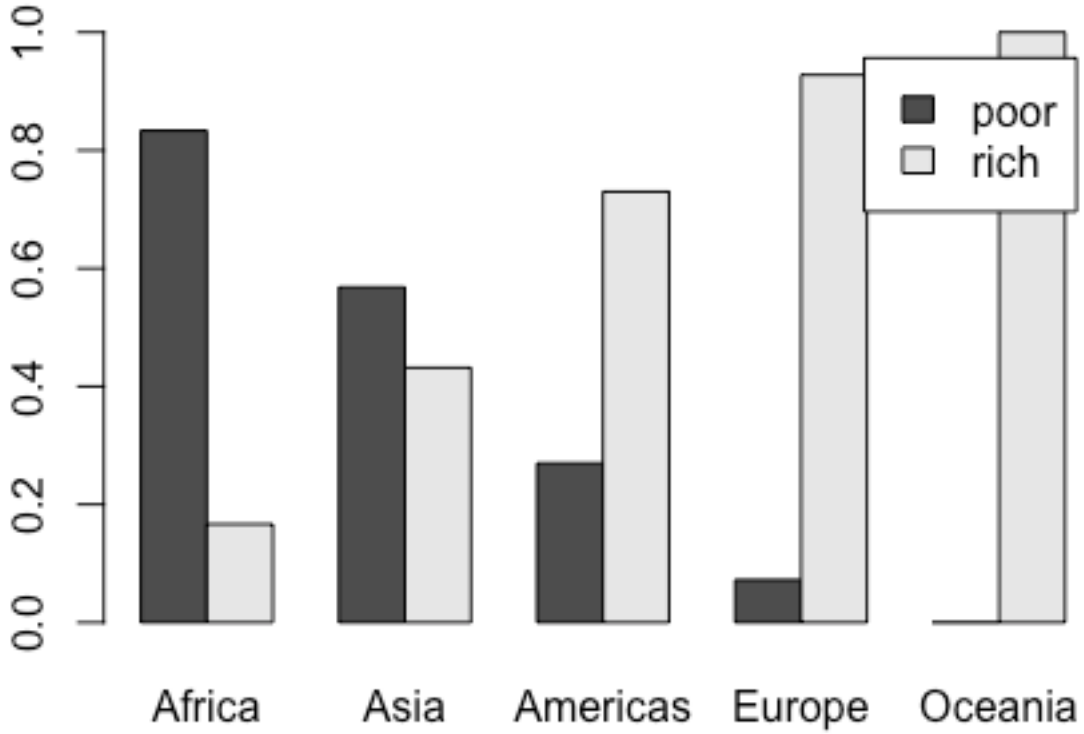
Some Arguments: Graphical Parameters

```
barplot(my.tab, beside=T, horiz=T)
```



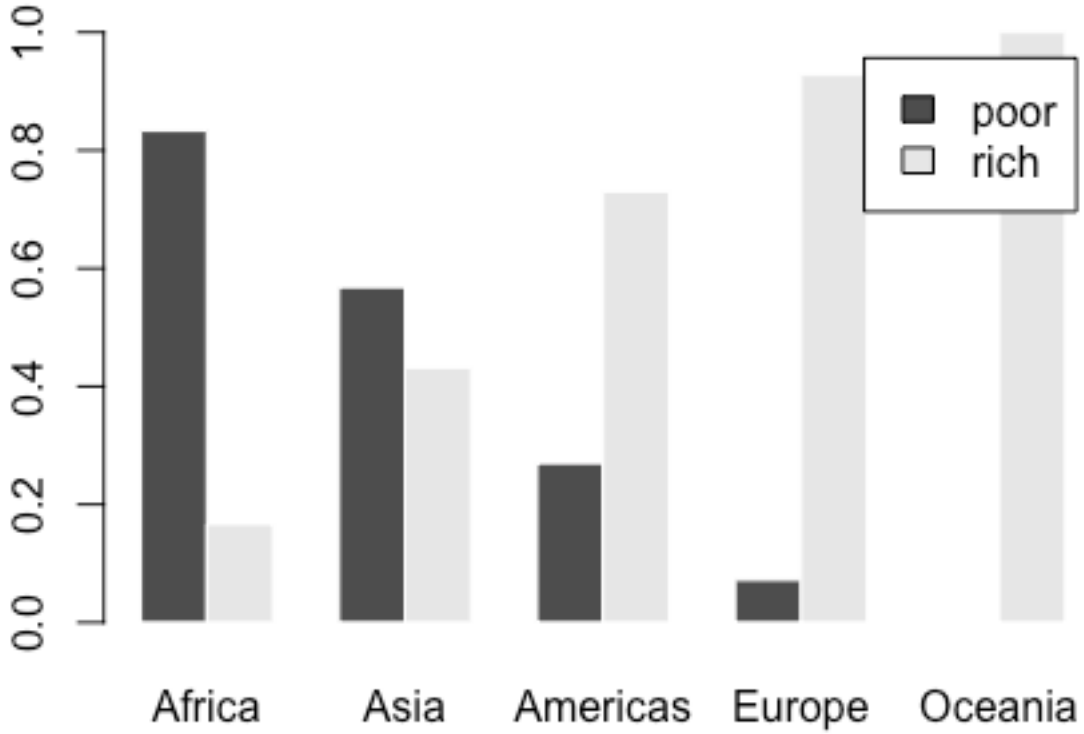
Some Arguments: Graphical Parameters

```
barplot(my.tab, beside=T, legend=T)
```



Some Arguments: Graphical Parameters

```
barplot(my.tab, beside=T, legend=T, border=F)
```



Use Help to Find More Arguments

```
?barplot()
```

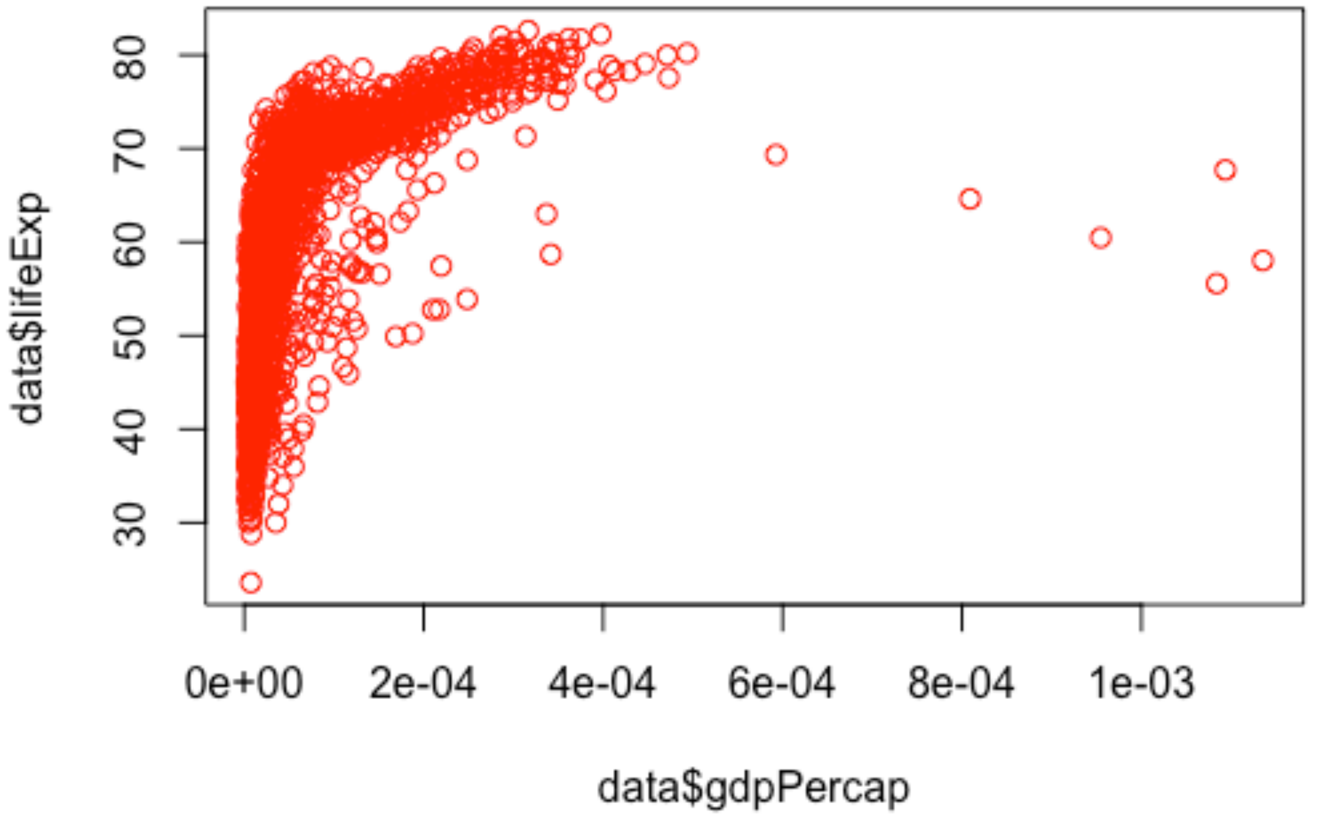
```
barplot(height, ...)

## Default S3 method:
barplot(height, width = 1, space = NULL,
        names.arg = NULL, legend.text = NULL, beside = FALSE,
        horiz = FALSE, density = NULL, angle = 45,
        col = NULL, border = par("fg"),
        main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
        xlim = NULL, ylim = NULL, xpd = TRUE, log = "",
        axes = TRUE, axisnames = TRUE,
        cex.axis = par("cex.axis"), cex.names = par("cex.axis"),
        inside = TRUE, plot = TRUE, axis.lty = 0, offset = 0,
        add = FALSE, args.legend = NULL, ...)
```

Arguments

`height`	either a vector or matrix of values describing the bars which make up the plot. If `height` is a vector, the plot consists of a sequence of rectangular bars with heights given by the values in the vector. If `height` is a matrix and `beside` is `FALSE` then each bar of the plot corresponds to a column of `height`, with the values in the column giving the heights of stacked sub-bars making up the bar. If `height` is a matrix and `beside` is `TRUE`, then the values in each column are juxtaposed rather than stacked.
`width`	optional vector of bar widths. Re-cycled to length the number of bars drawn. Specifying a single value will have no visible effect unless `xlim` is specified.

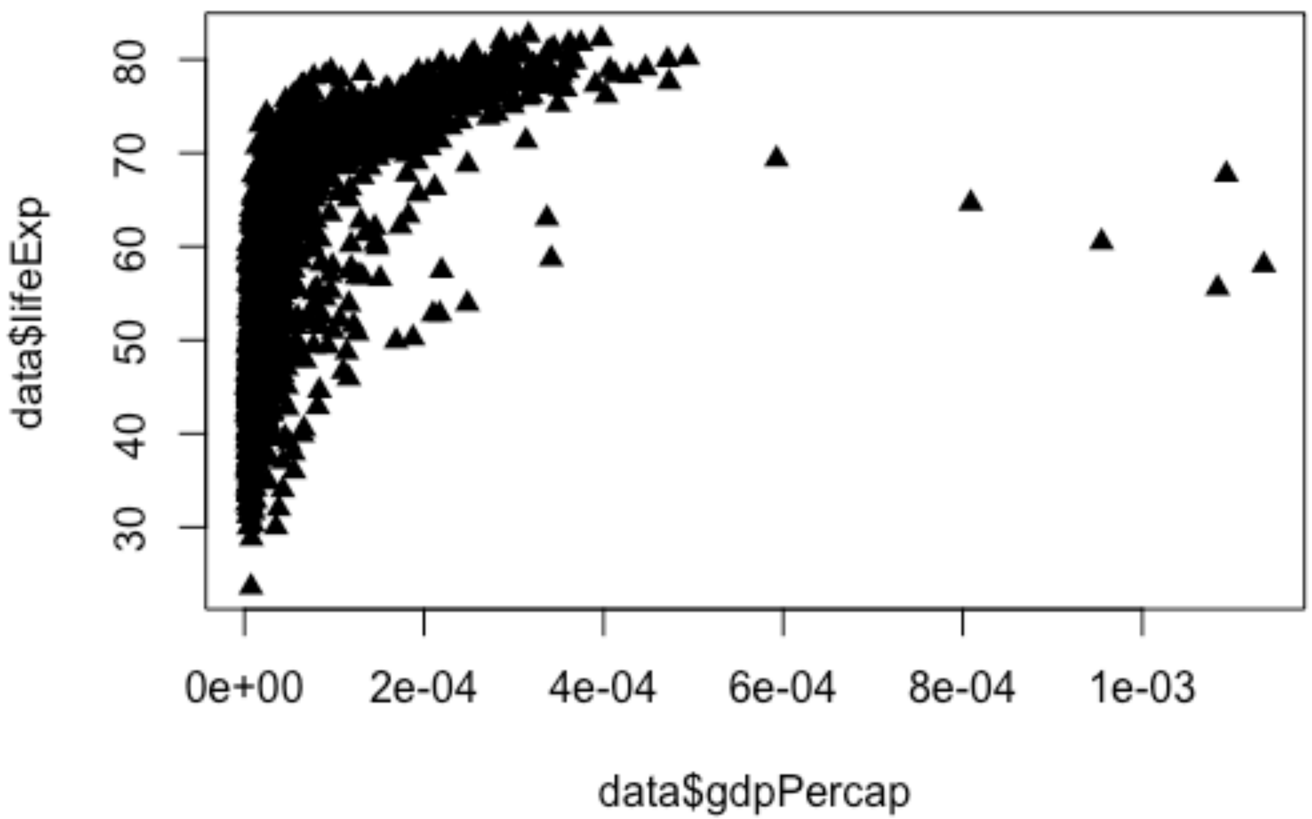
Some Arguments that Work for Most High-level Functions

```
plot(data$gdpPercap, data$lifeExp, col="red")
```



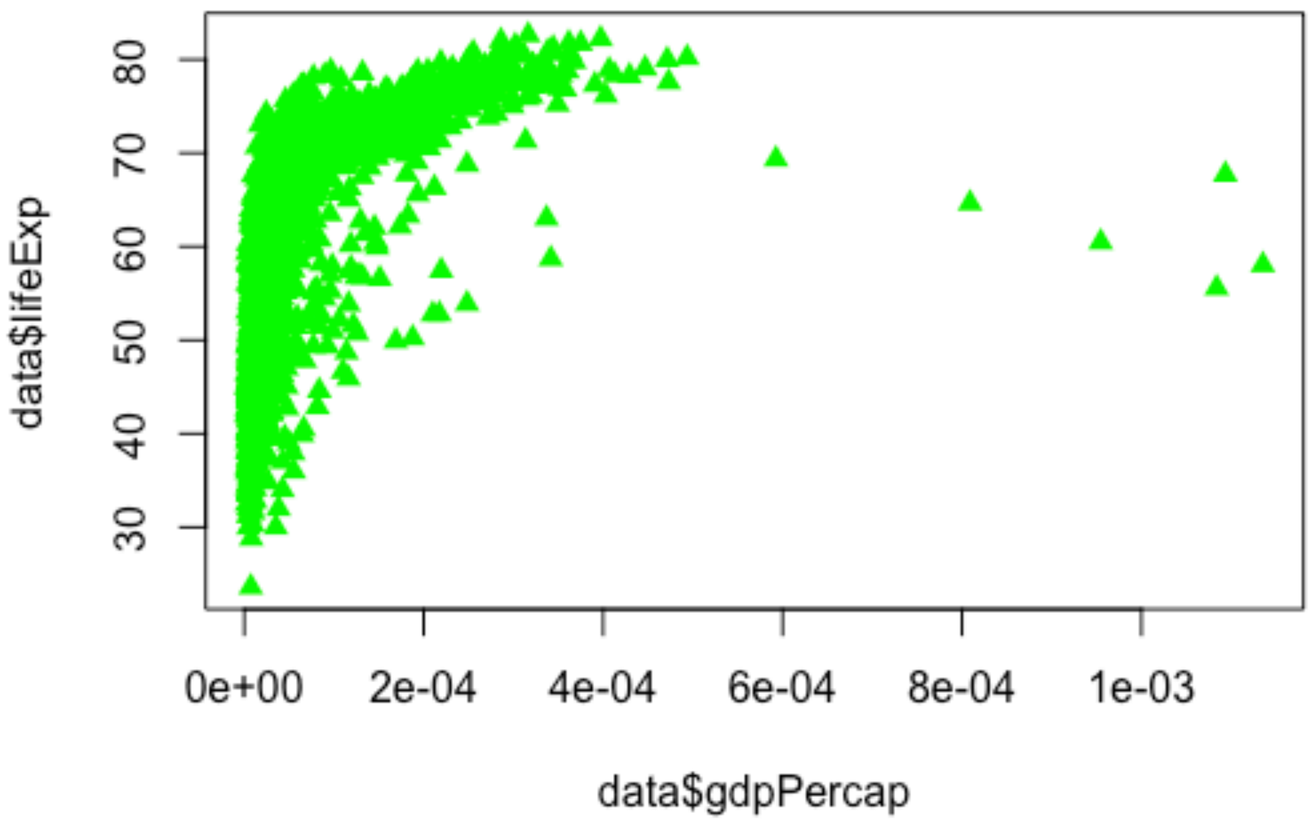
Some Arguments that Work for Most High-level Functions

```
plot(data$gdpPercap, data$lifeExp,  pch=17)
```



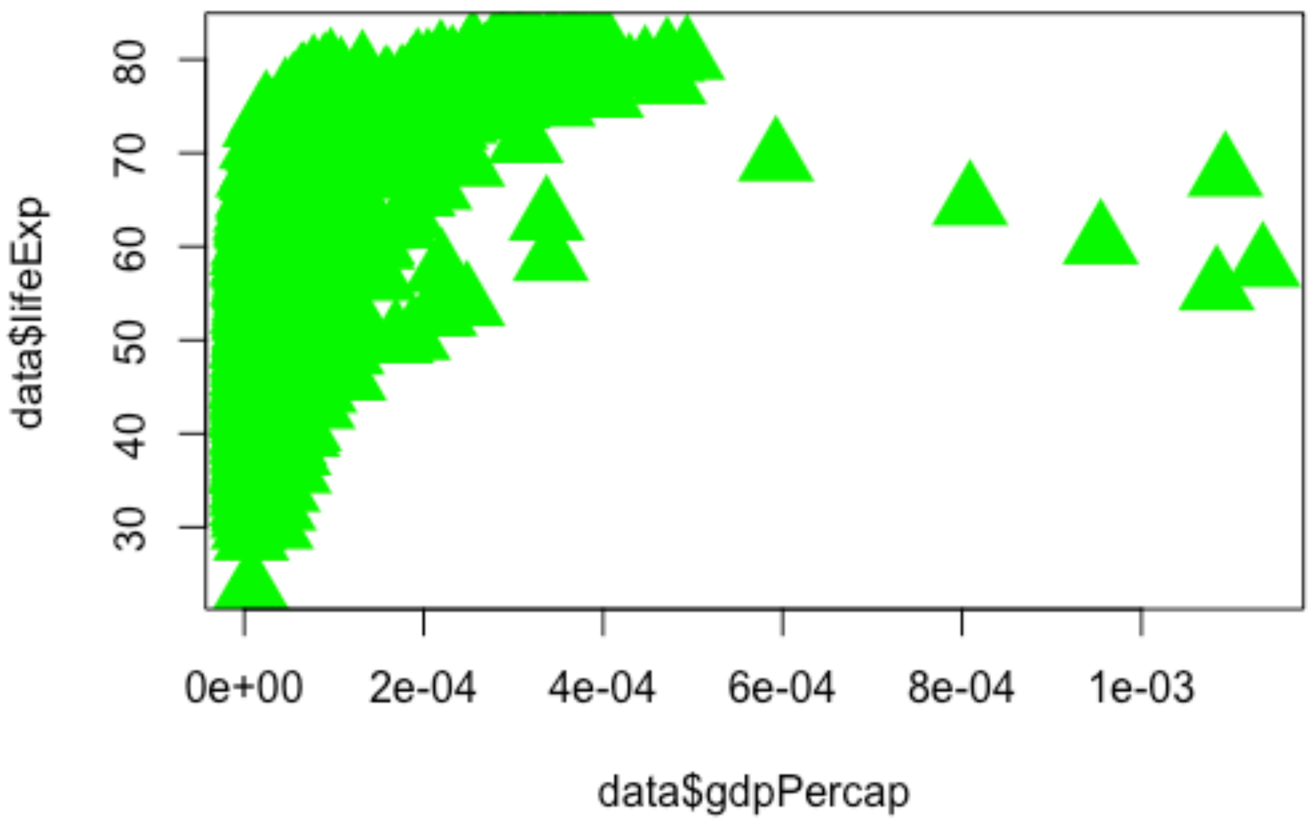
Some Arguments that Work for Most High-level Functions

```
plot(data$gdpPercap, data$lifeExp, pch=17, col="green")
```

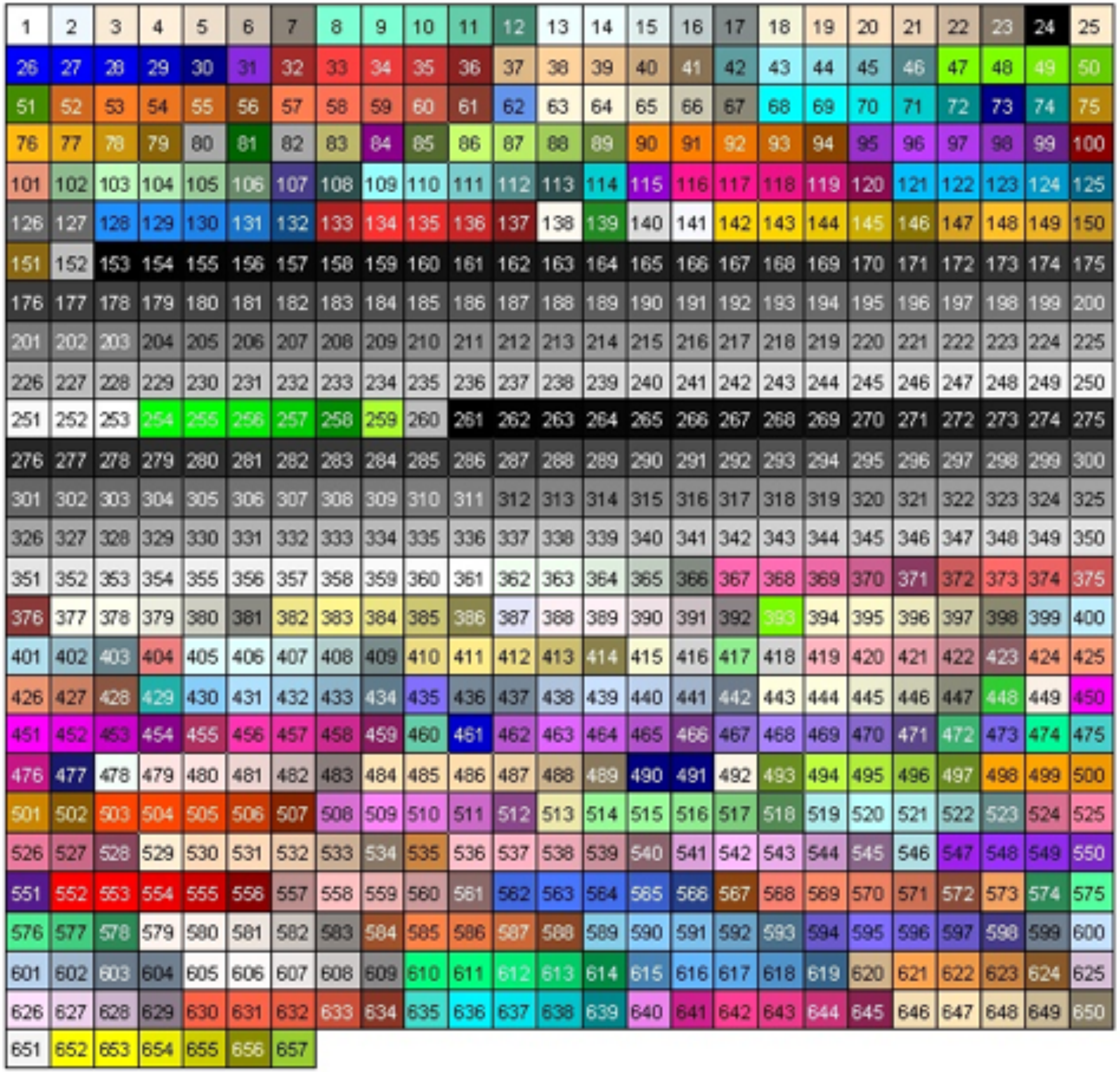


Some Arguments that Work for Most High-level Functions

```
plot(data$gdpPercap, data$lifeExp, pch=17, col="green", cex=3)
```



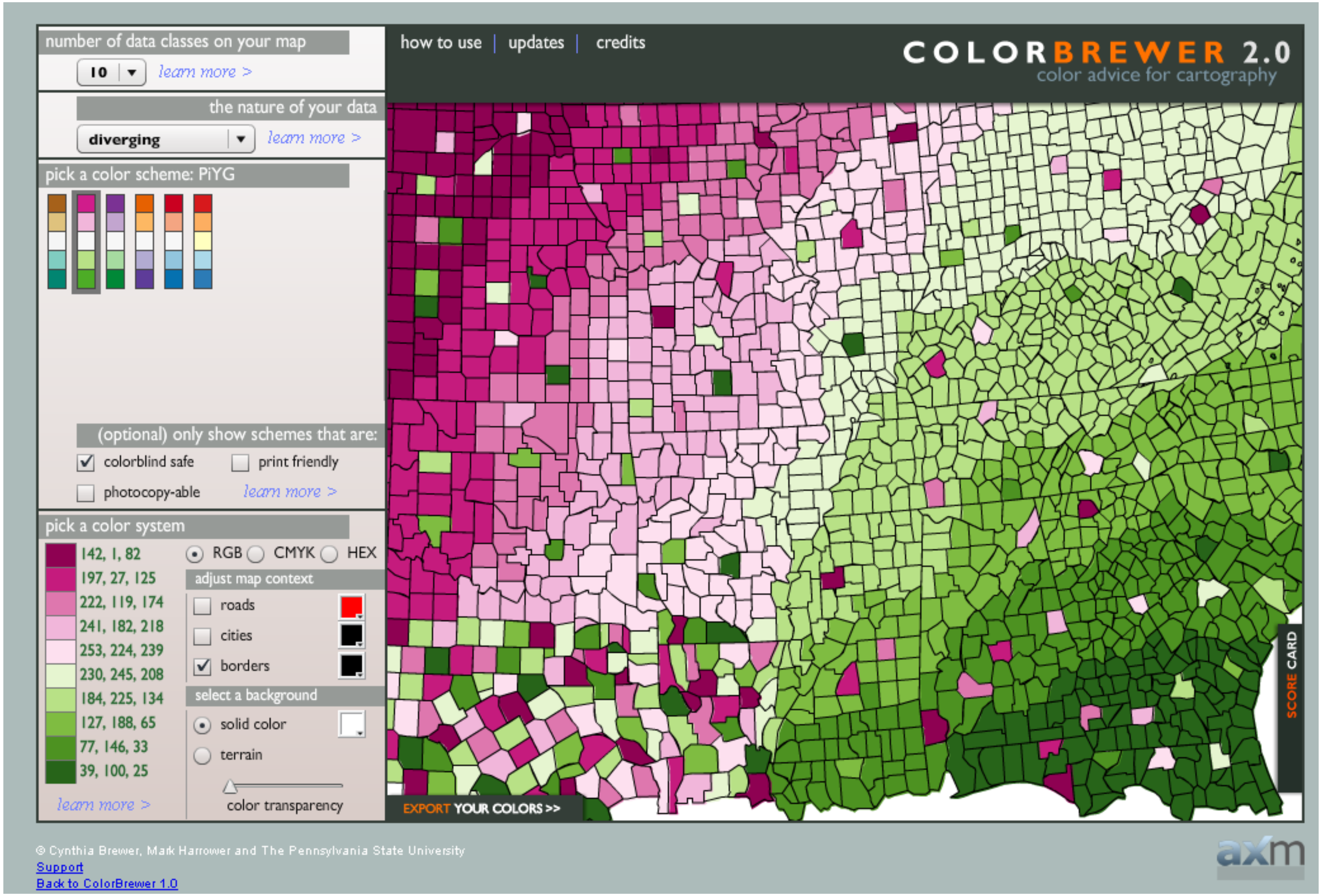
R colors


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425
426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475
476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500
501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525
526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550
551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575
576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600
601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625
626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650
651 652 653 654 655 656 657

How to find your favorite color

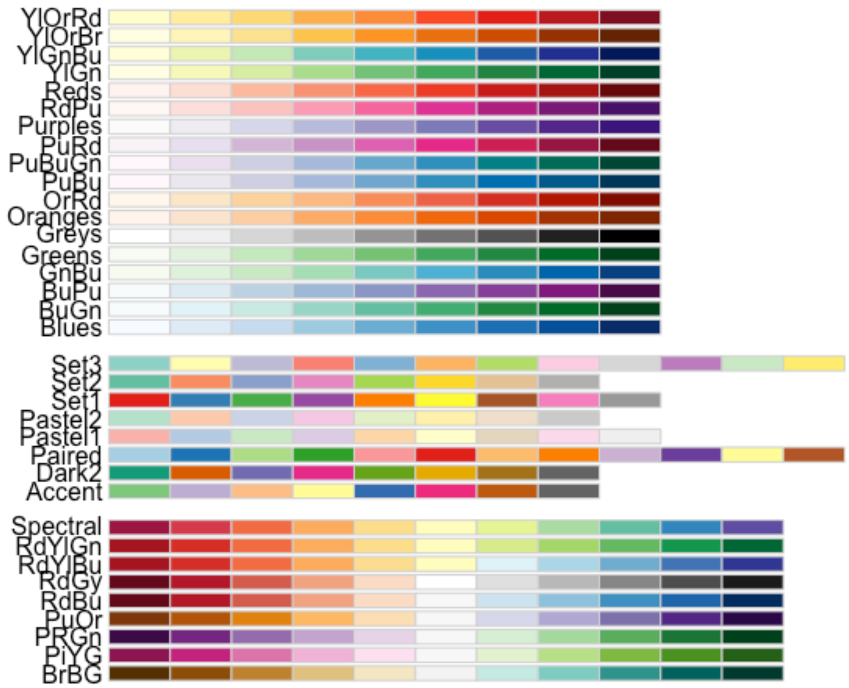
```
colors()[c(1, 280, 637)]

[1] "white" "grey19" "turquoise2"
```

colorbrewer2.org



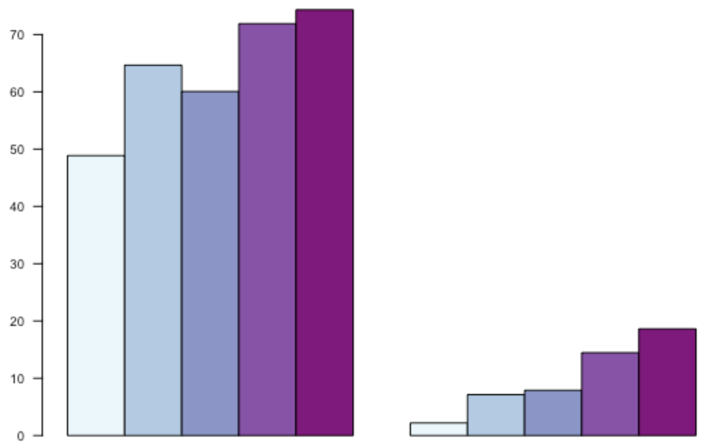
Using a Color Brewer pallette

```
library(RColorBrewer)

display.brewer.all()
```



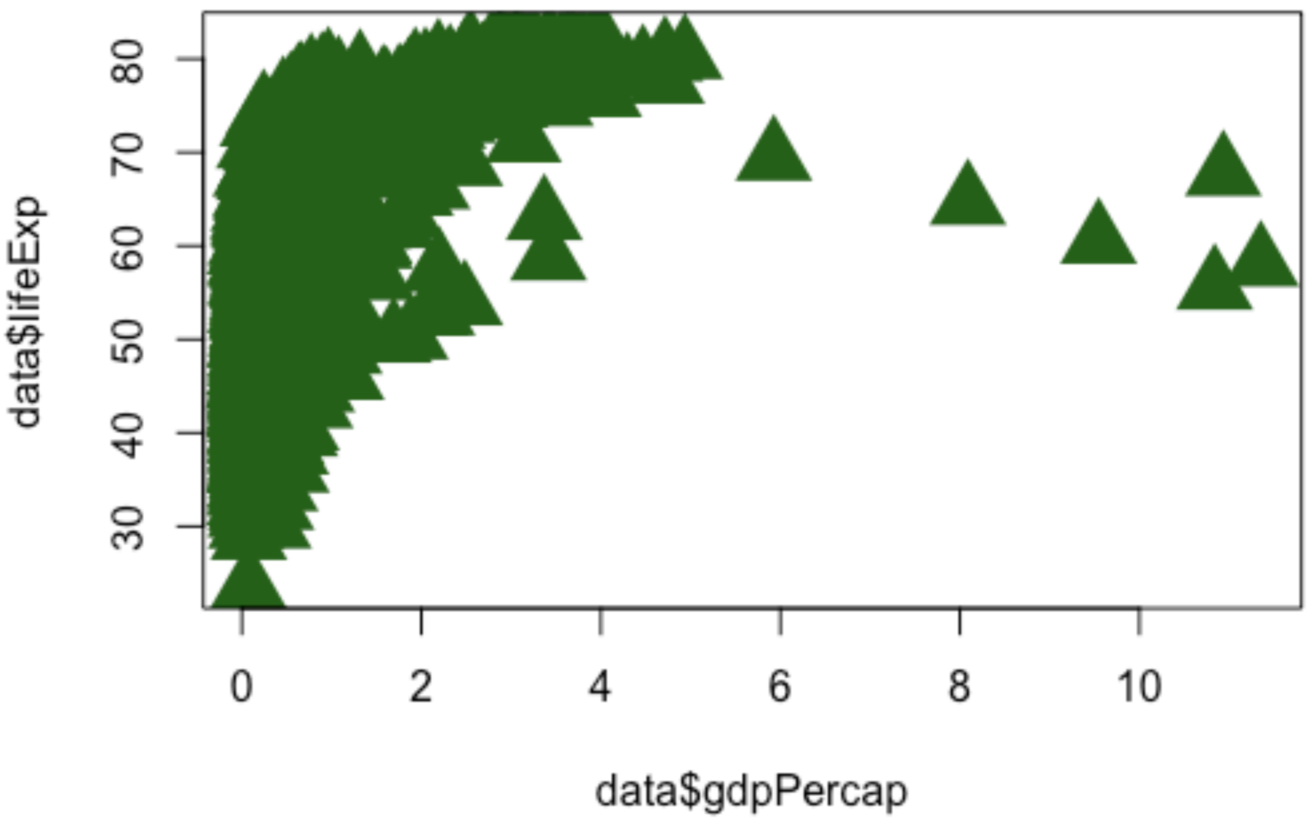
Using a Color Brewer pallette

```
my.colors <- brewer.pal(5, "BuPu")

barplot(cbind(life$x, gdp$x), beside=T, col=my.colors)
```

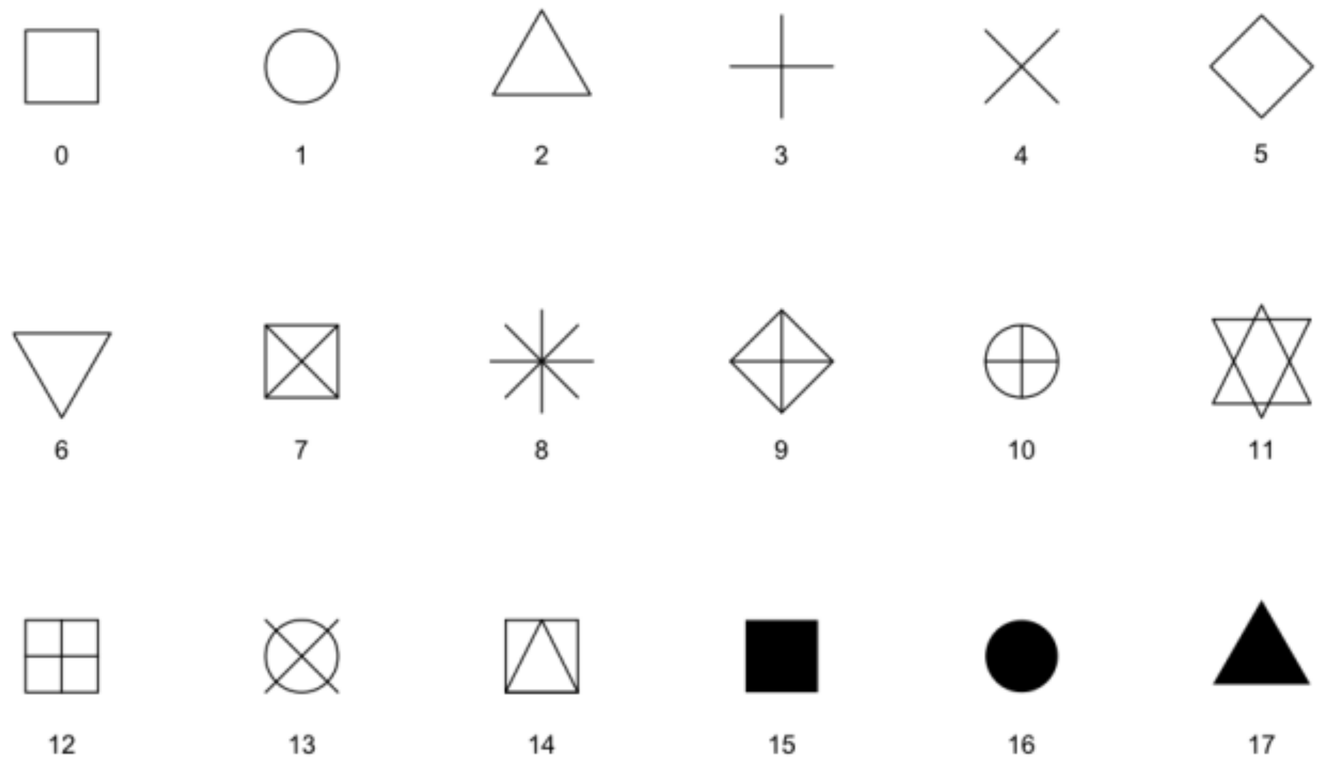


How to find your favorite color

```
plot(data$gdpPercap, data$lifeExp, pch=17, col=rgb(39, 100, 25,
max=255), cex=3)
```

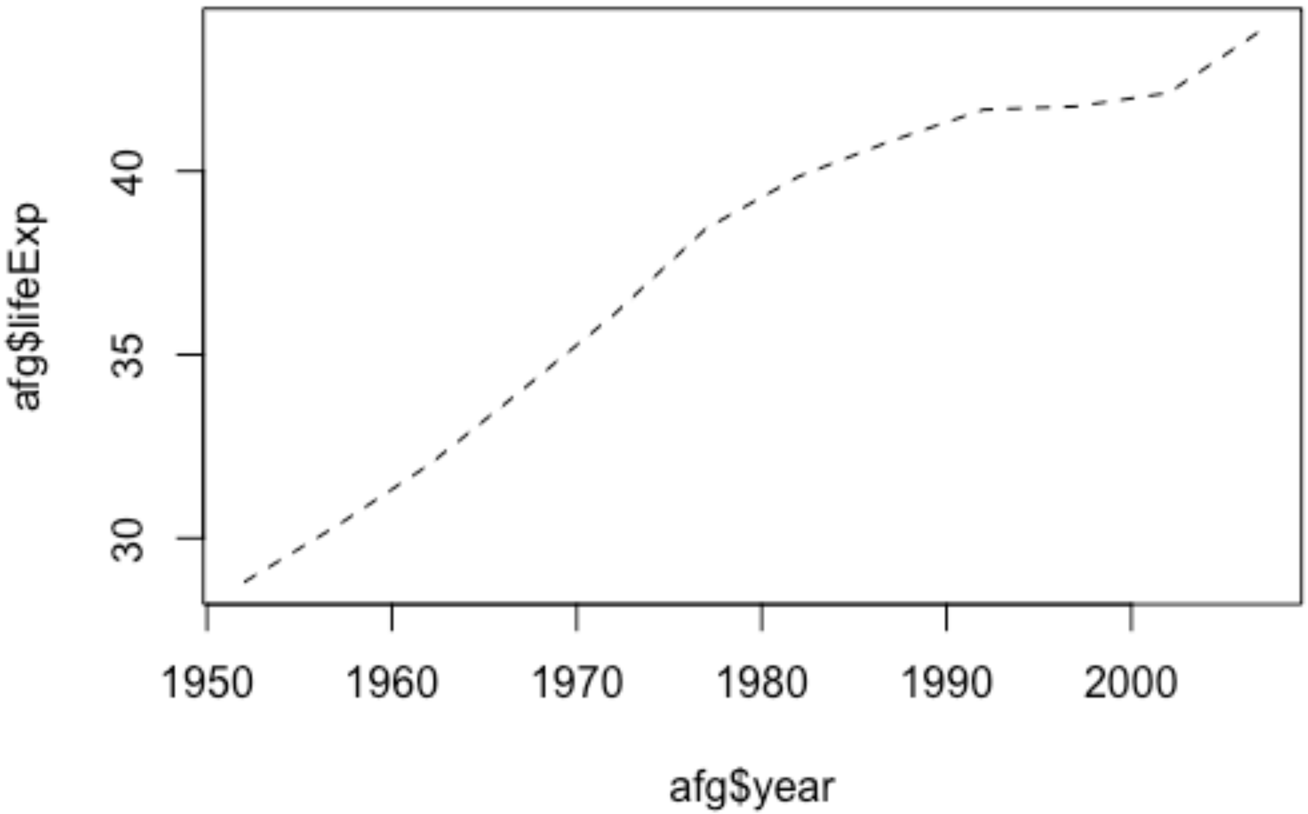


Plotting Characters



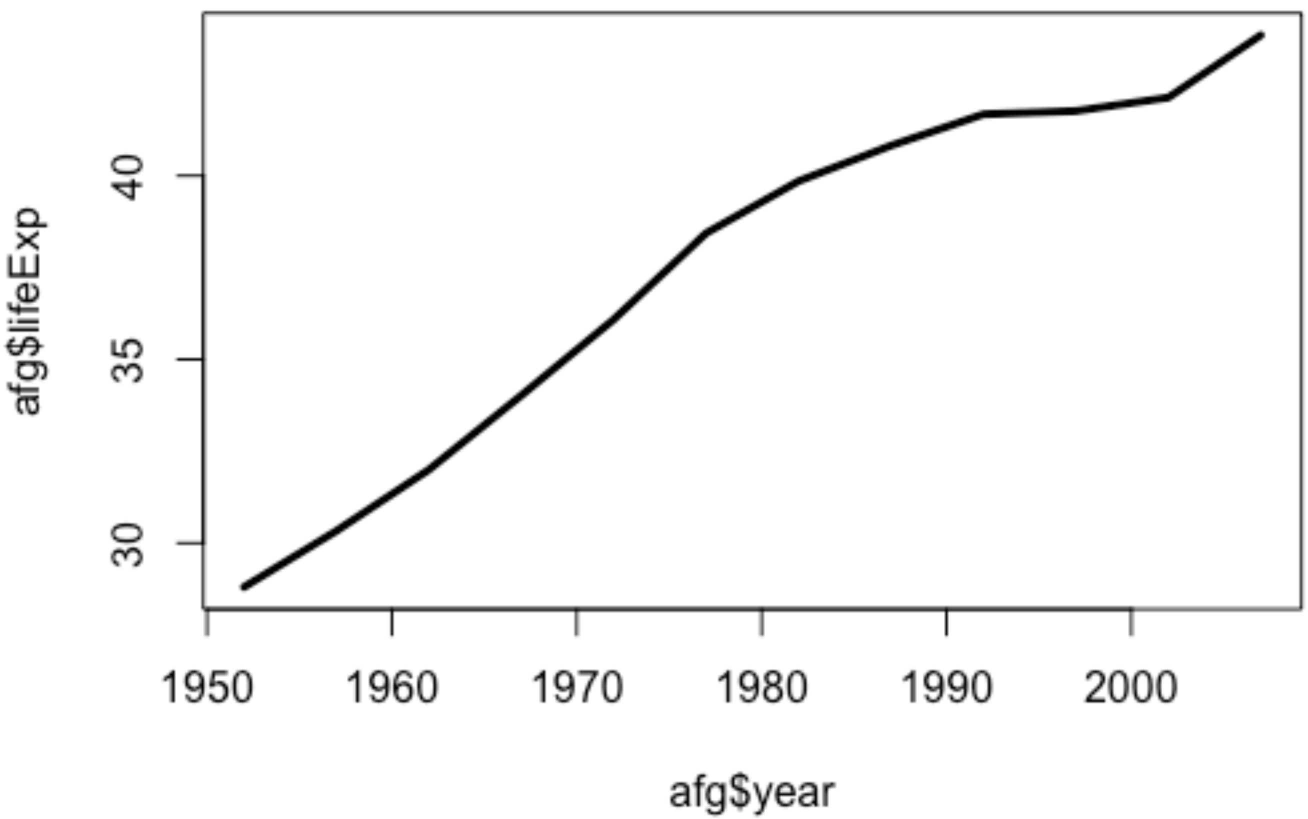
Some Arguments that Work for Most High-level Functions

```
plot(afg$year, afg$lifeExp, type="l", lty="dashed")
```



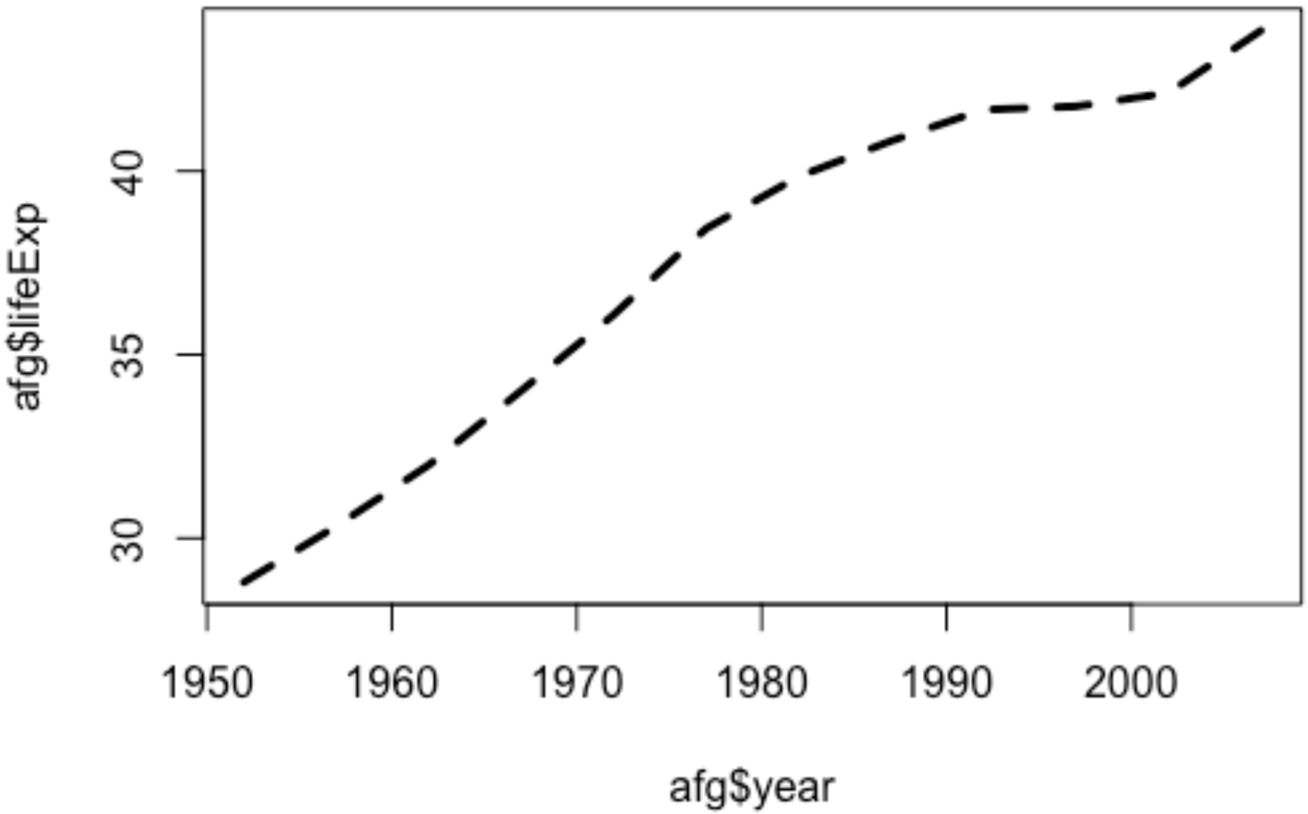
Some Arguments that Work for Most High-level Functions

```
plot(afg$year, afg$lifeExp, type="l", lwd=3)
```



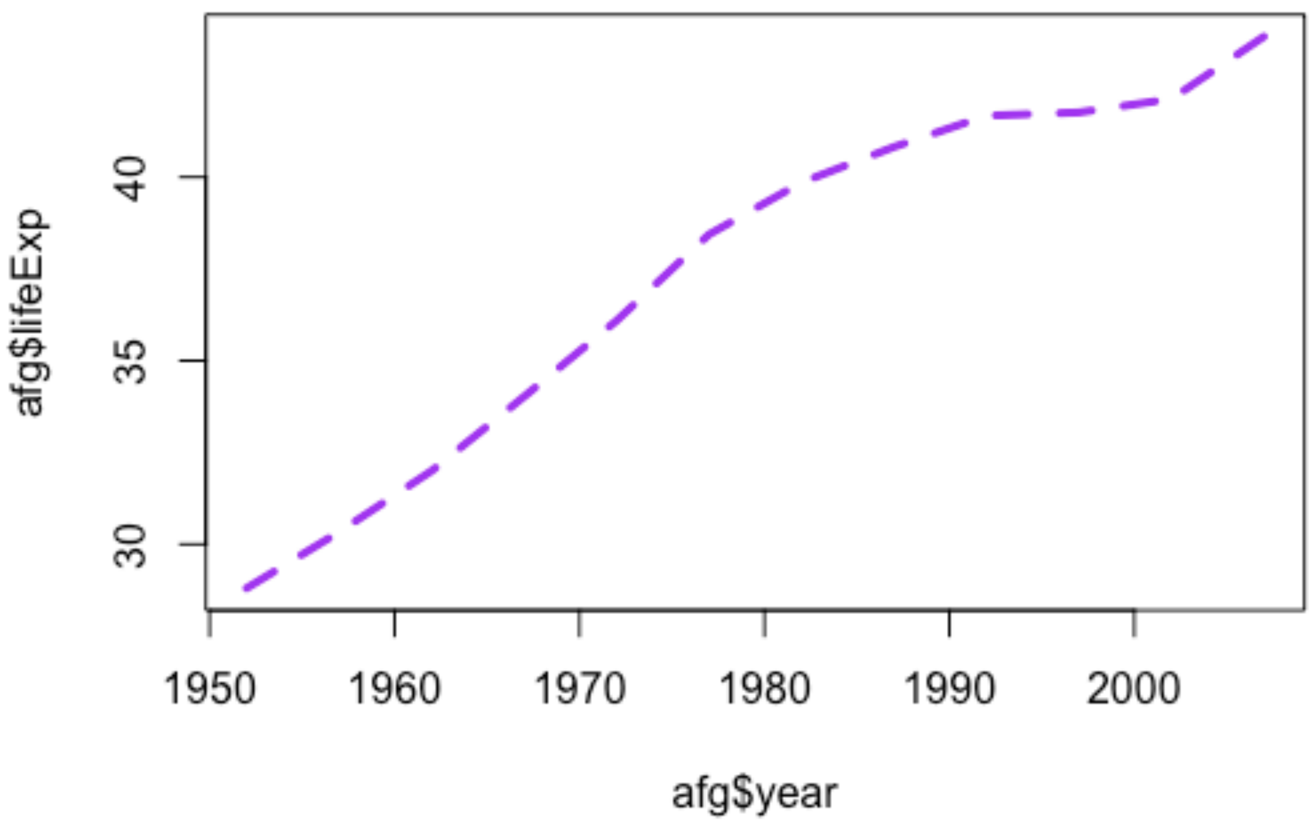
Some Arguments that Work for Most High-level Functions

```
plot(afg$year, afg$lifeExp, type="l", lwd=3, lty="dashed")
```



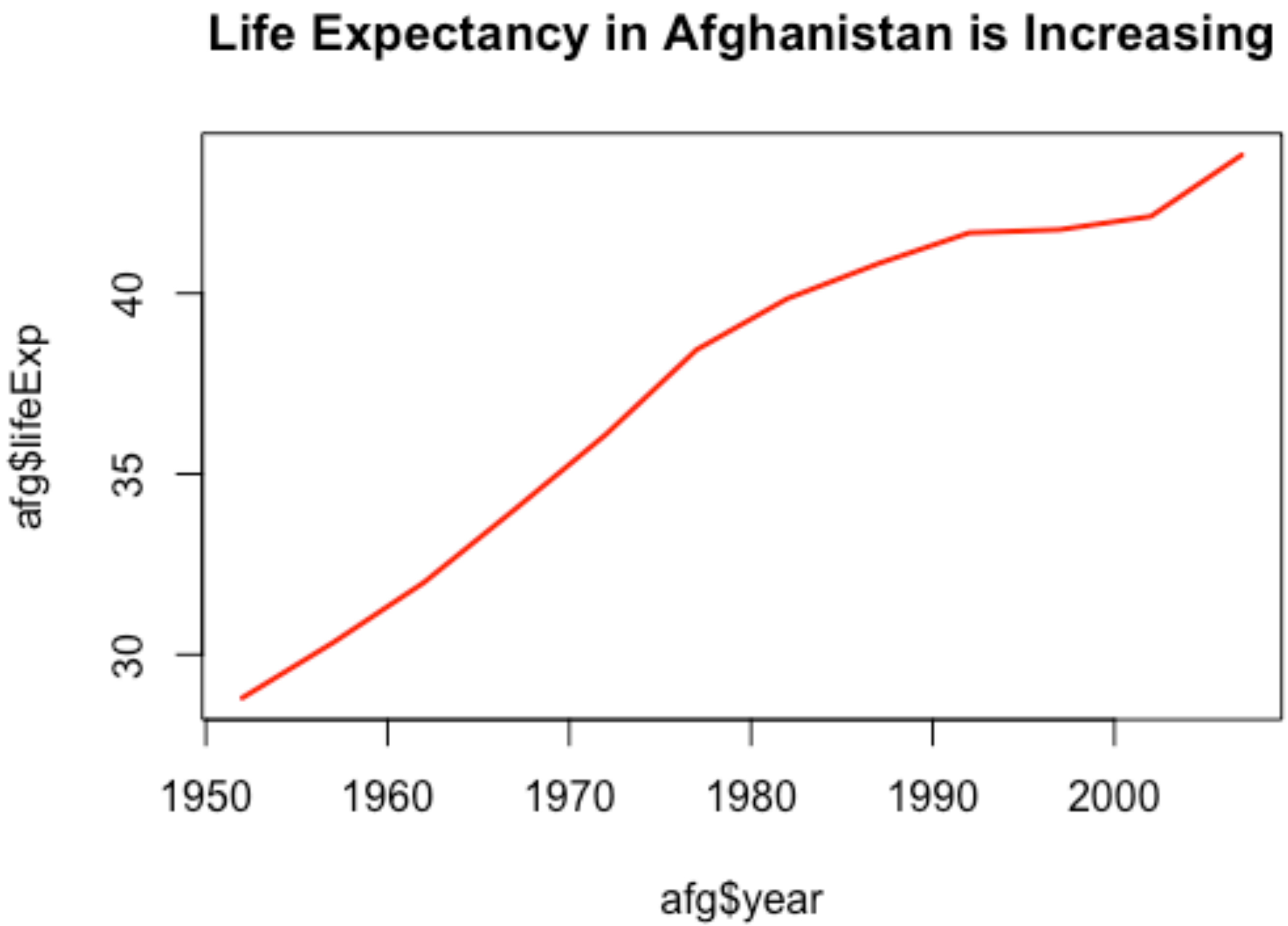
Some Arguments that Work for Most High-level Functions

```
plot(afg$year, afg$lifeExp, type="l", lwd=3, lty="dashed",
col="purple")
```



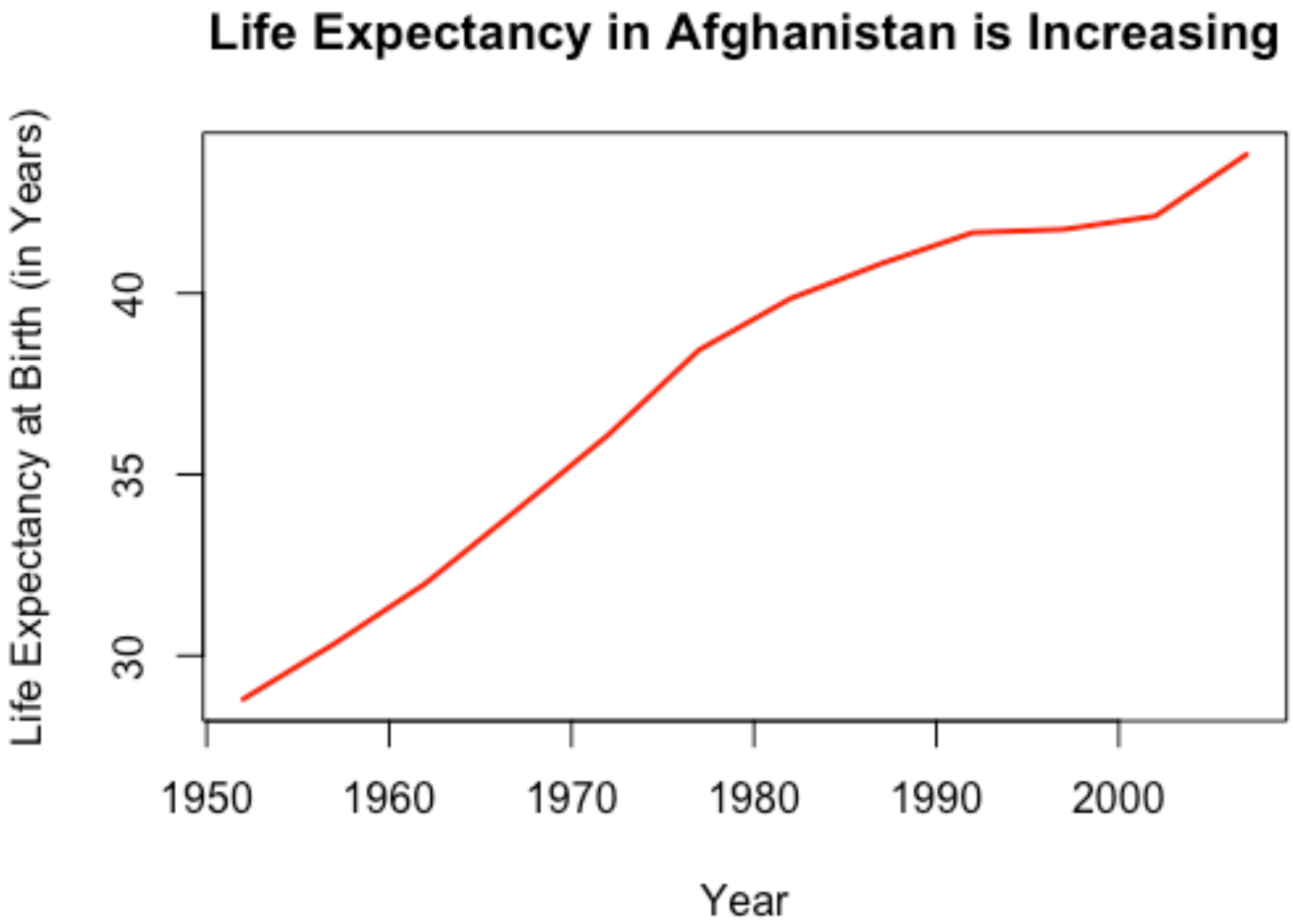
More Arguments: Title, Labels, Axes

```
plot(afg$year, afg$lifeExp, type="l", lwd=2, col="red",
main="Life Expectancy in Afghanistan is Increasing")
```



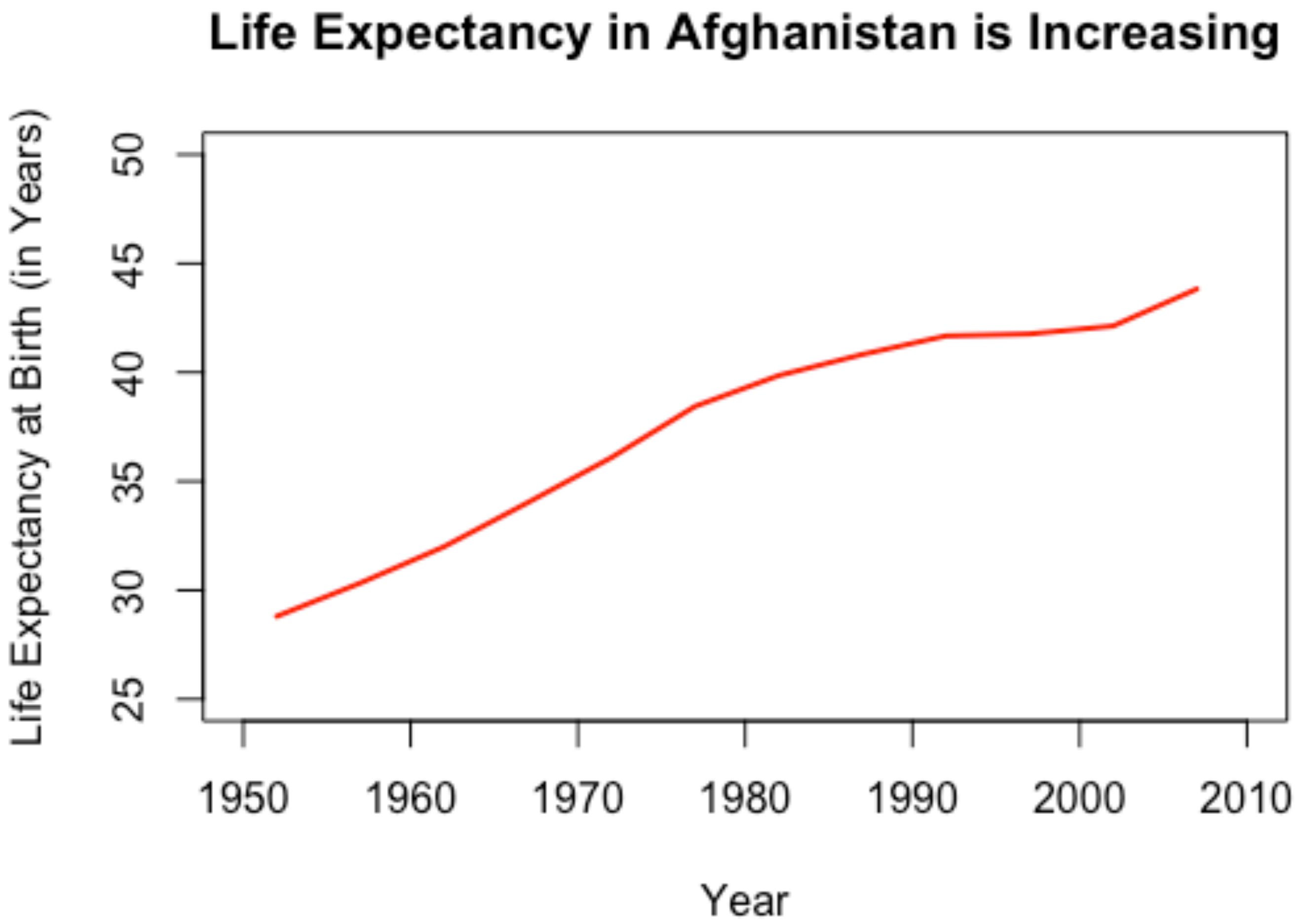
More Arguments: Title, Labels, Axes

```
plot(afg$year, afg$lifeExp, type="l", lwd=2, col="red",
main="Life Expectancy in Afghanistan is Increasing",
xlab="Year", ylab="Life Expectancy at Birth (in Years)")
```



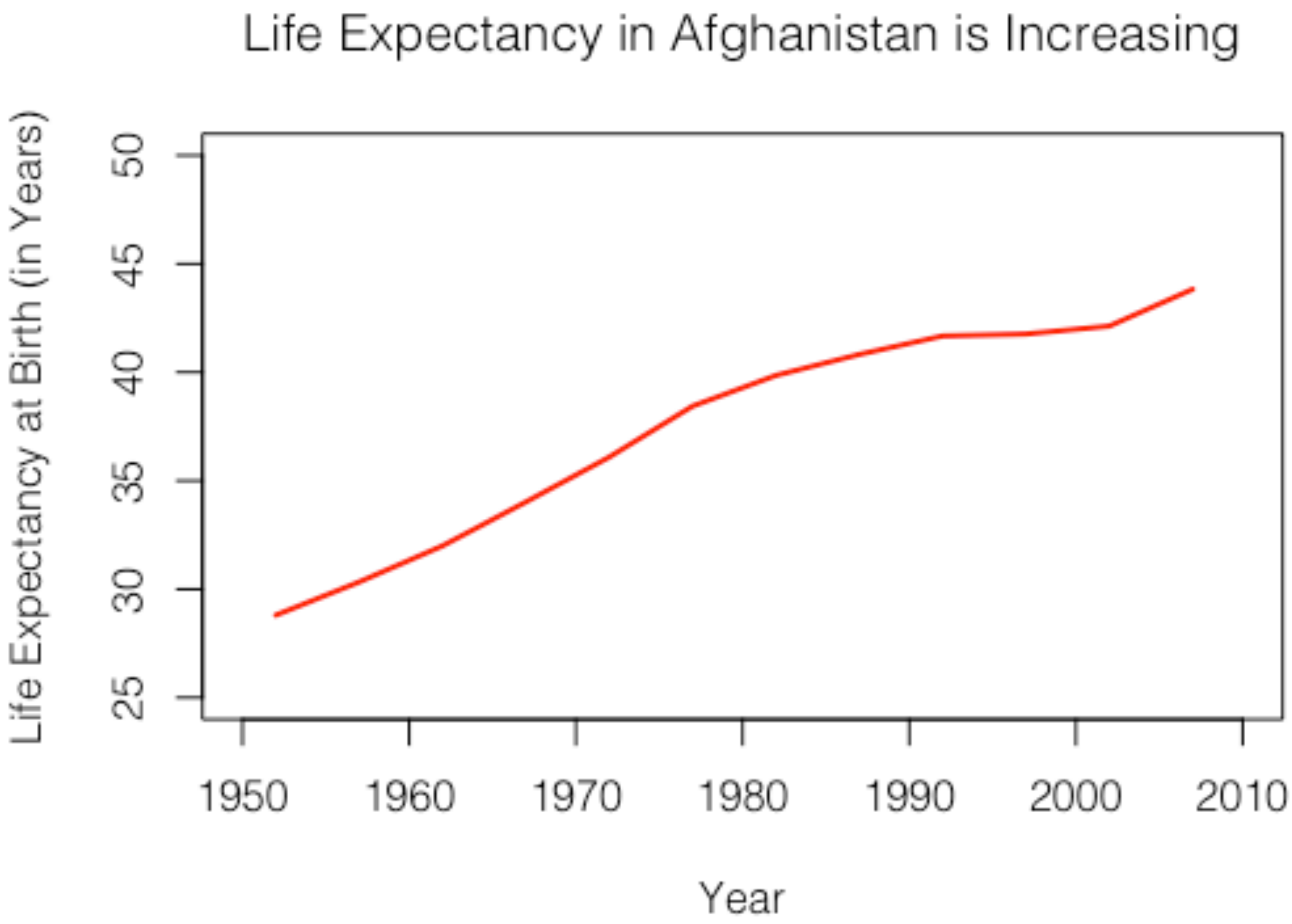
More Arguments: Title, Labels, Axes

```
plot(afg$year, afg$lifeExp, type="l", lwd=2, col="red",… ,
ylim=c(25, 50), xlim=c(1950, 2010))
```



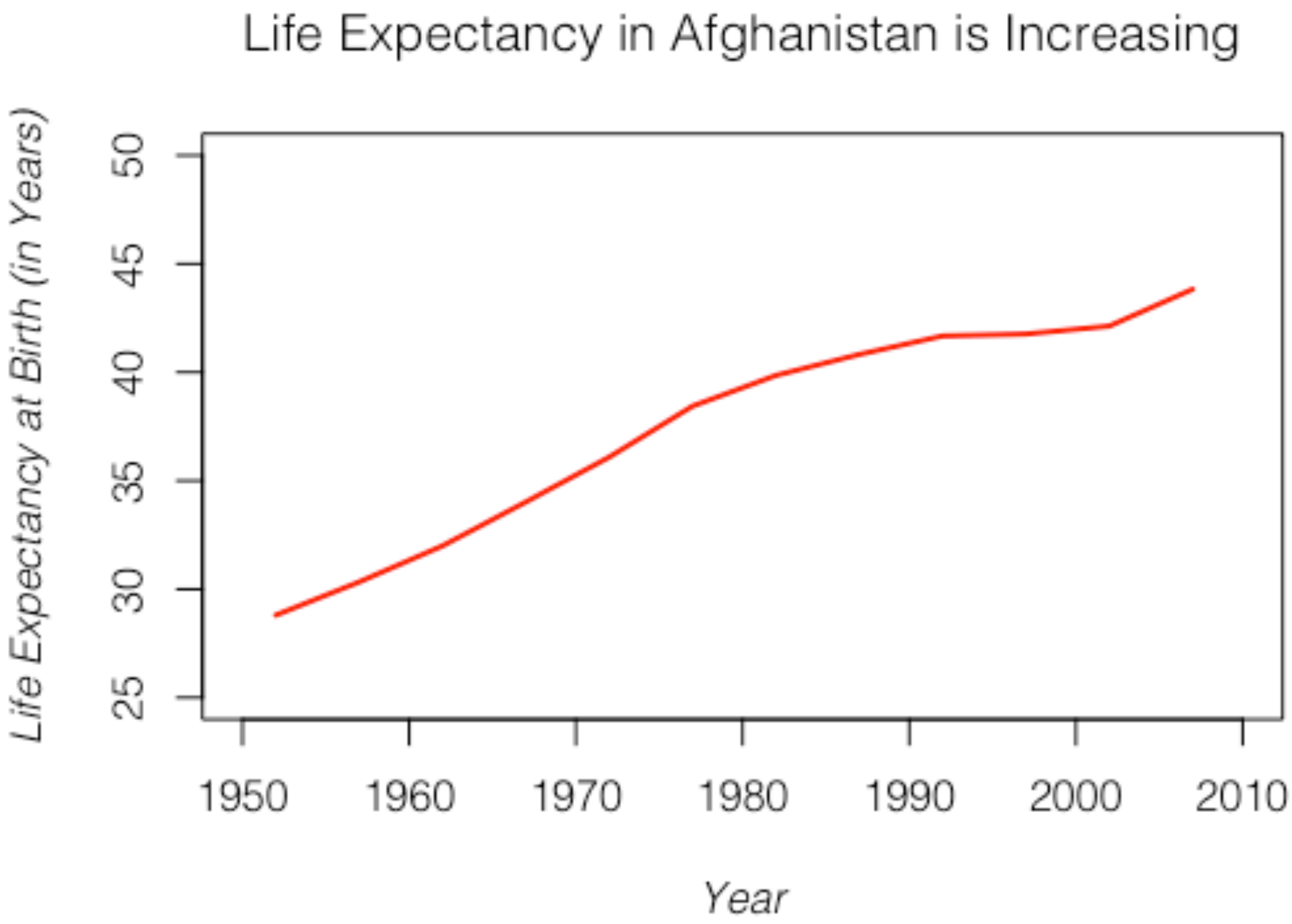
More Arguments: Title, Labels, Axes

```
plot(afg$year, afg$lifeExp, type="l", lwd=2, col="red", … ,
family="Helvetica Light")
```



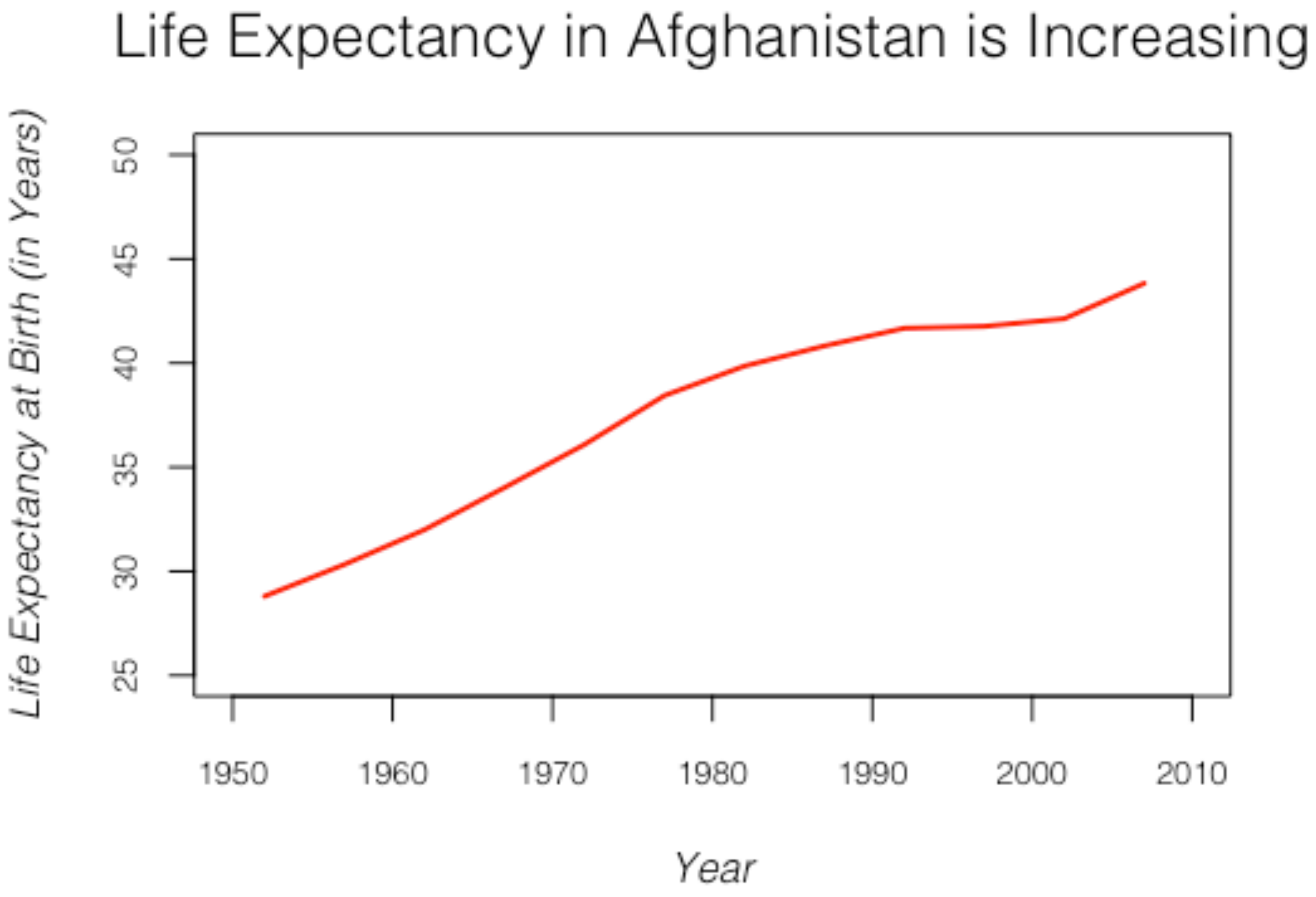
More Arguments: Title, Labels, Axes

```
plot(afg$year, afg$lifeExp, type="l", lwd=2, col="red", … ,
family="Helvetica Light", font.main=1, font.lab=3)
```



More Arguments: Title, Labels, Axes

```
plot(afg$year, afg$lifeExp, type="l", lwd=2, col="red", … ,
family="Helvetica Light", font.main=1, font.lab=3, cex.main=1.5,
cex.lab=1, cex.axis=.8)
```



The `par()` Function

Many arguments can be applied directly to high level functions such as `plot()`

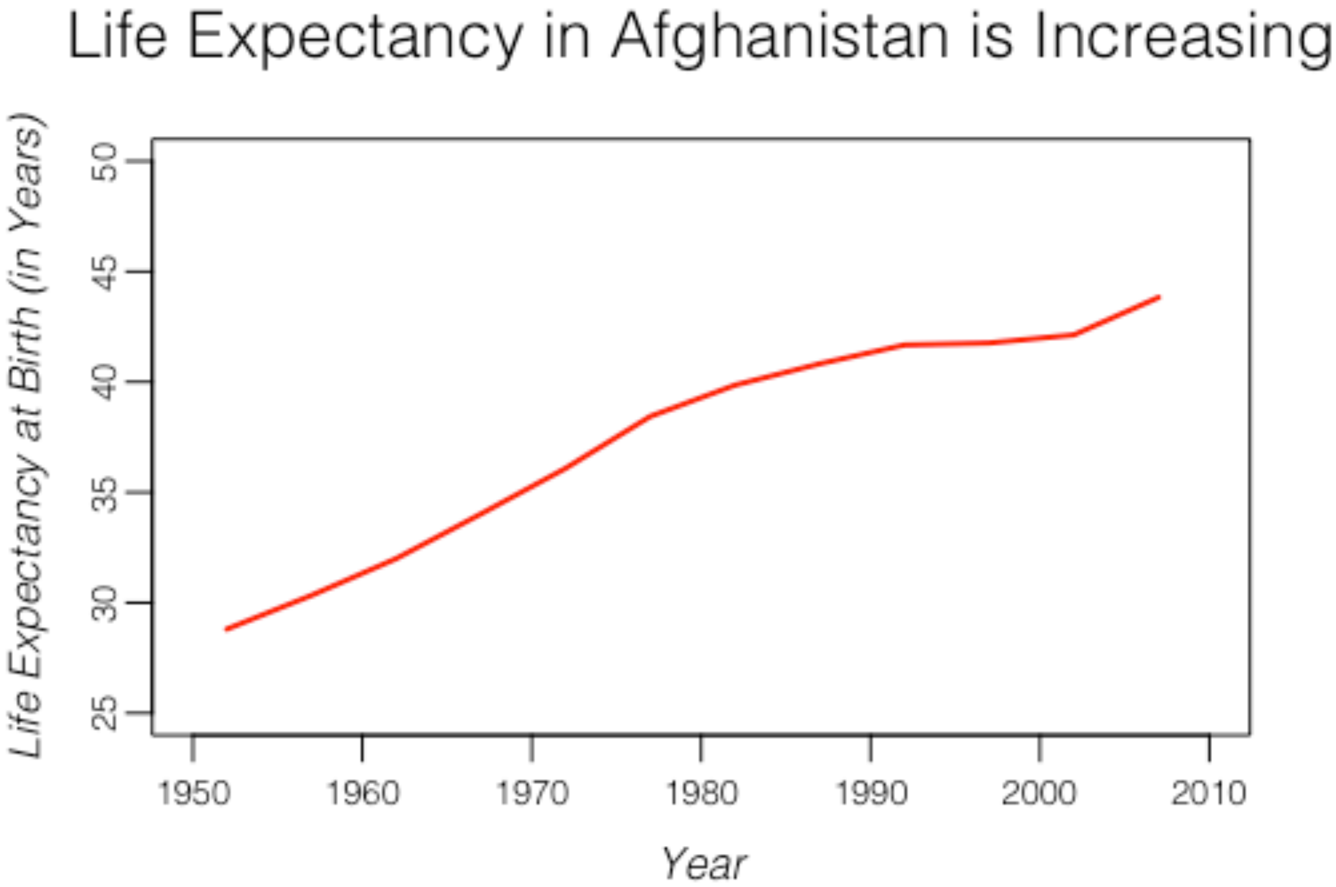
See help for all possible arguments and how to define them: `?plot()`

Arguments can also be specified using the `par()` function

This sets parameters permanently for a session and applies them to all plots

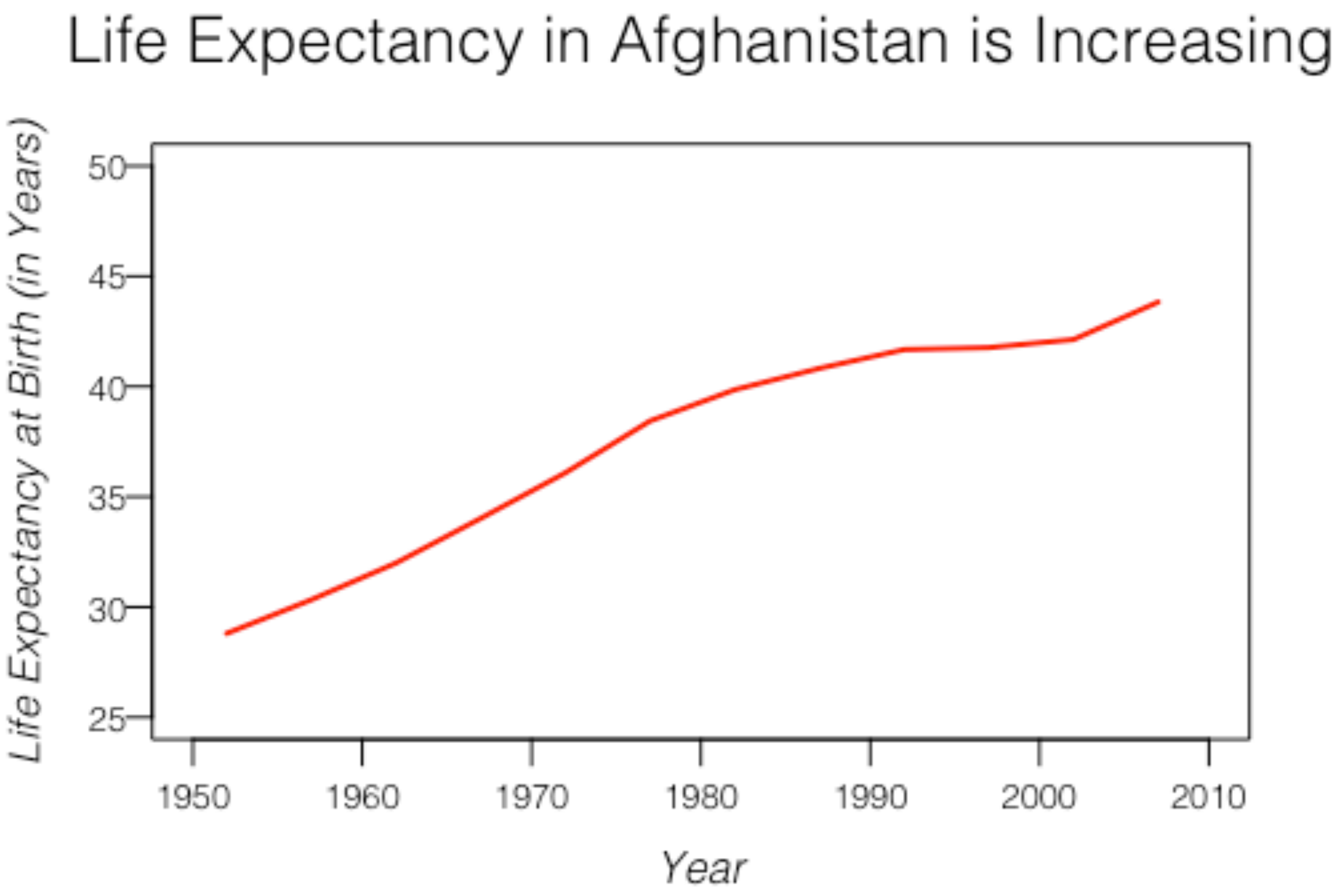
Some more arguments with `par()`

```
par(mgp=c(2, .5, 0))
```



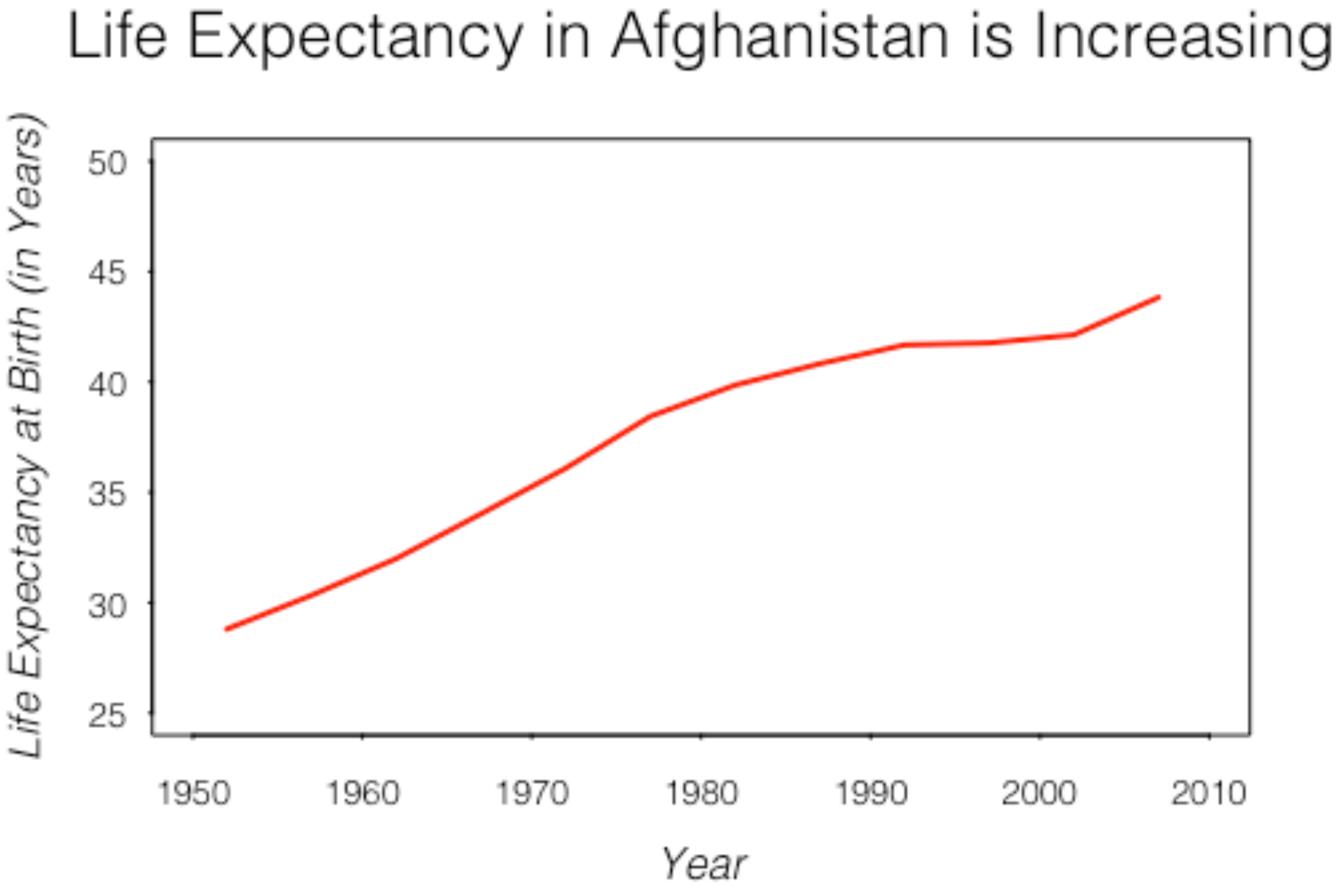
Some more arguments with `par()`

```
par(mgp=c(2, .5, 0), las=1)
```



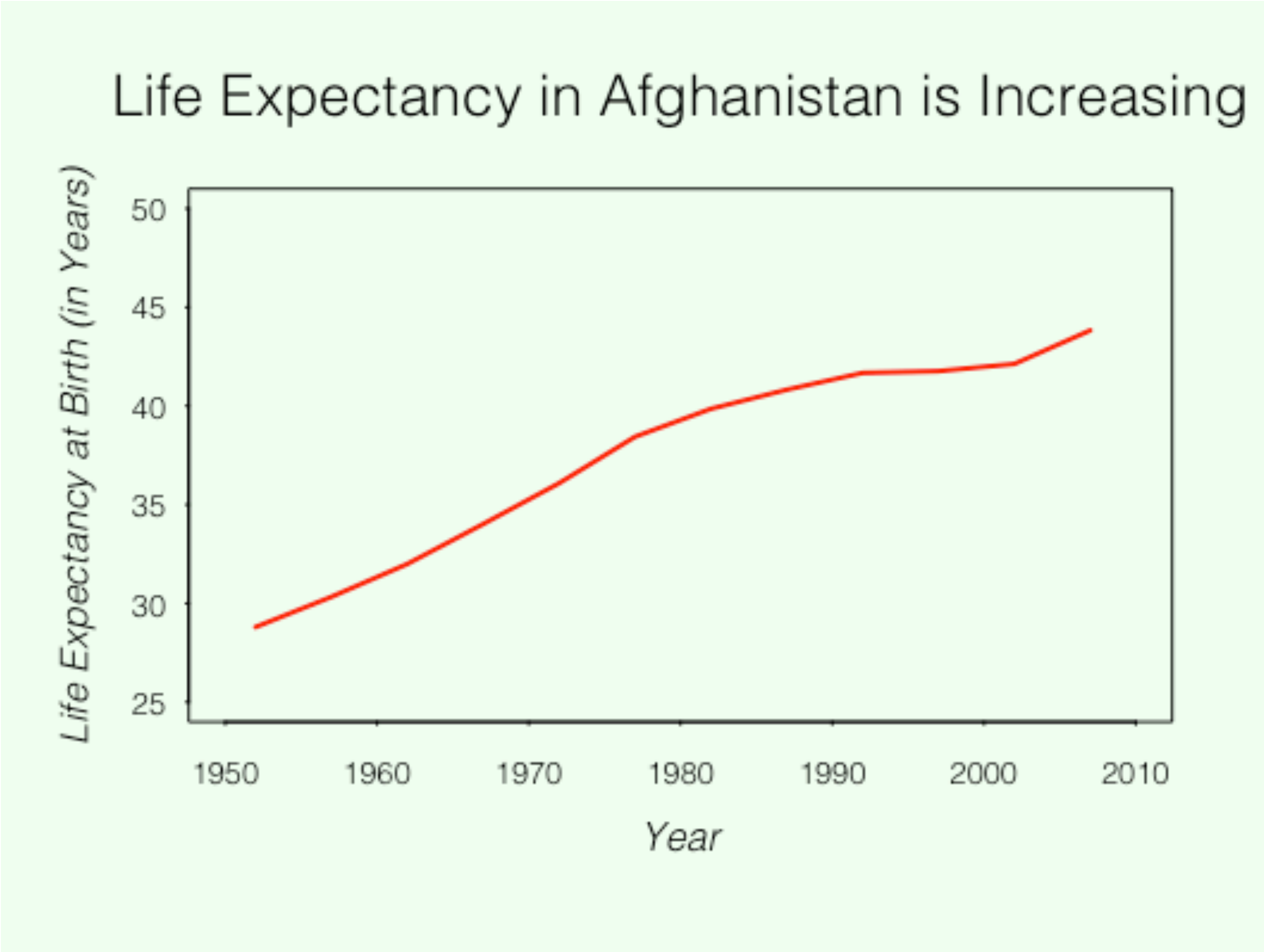
Some more arguments with `par()`

```
par(mgp=c(2, .5, 0), las=1, tck=-0.005)
```



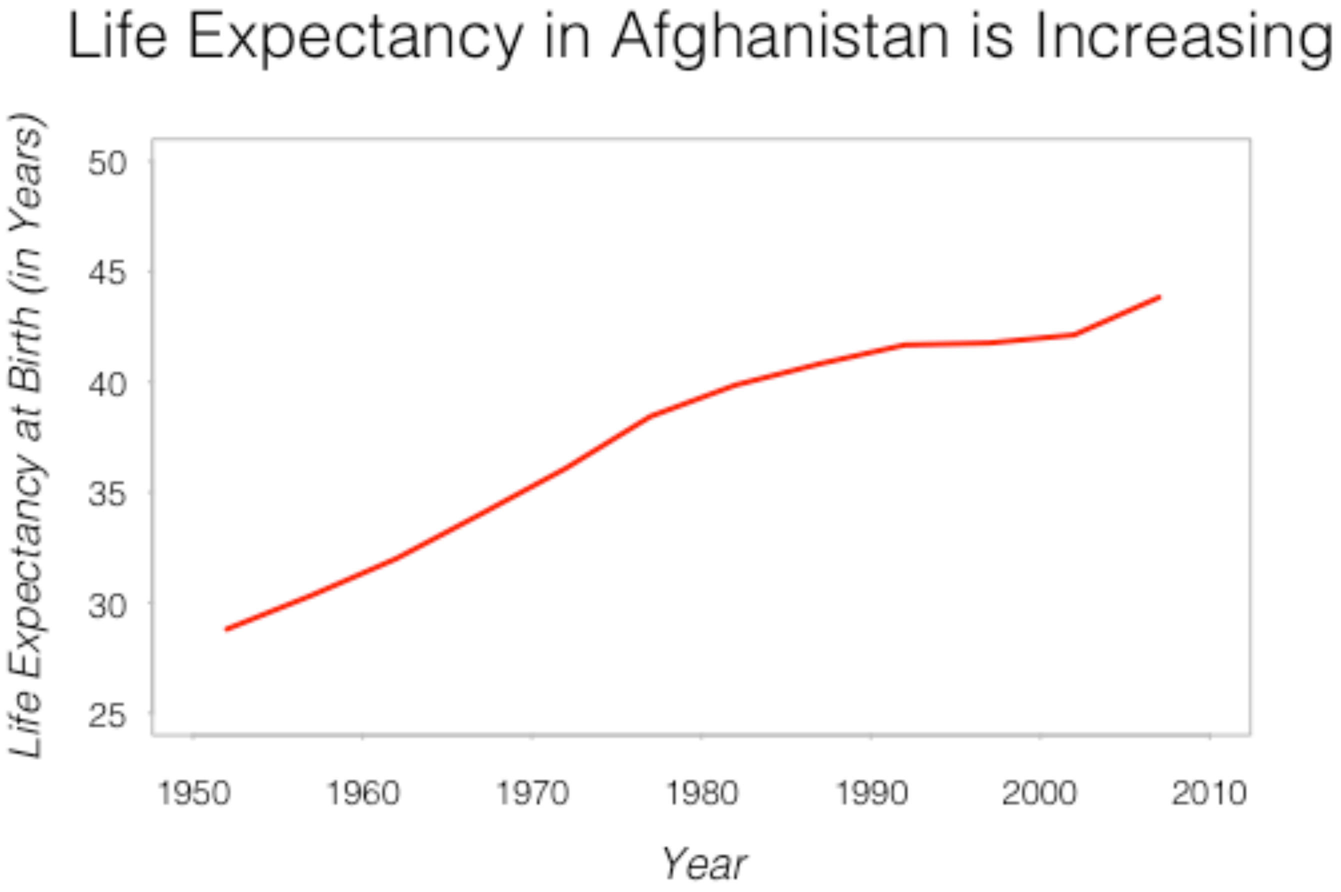
Some more arguments with `par()`

```
par(mgp=c(2, .5, 0), las=1, tck=-0.005, bg="honeydew")
```



Some more arguments with `par()`

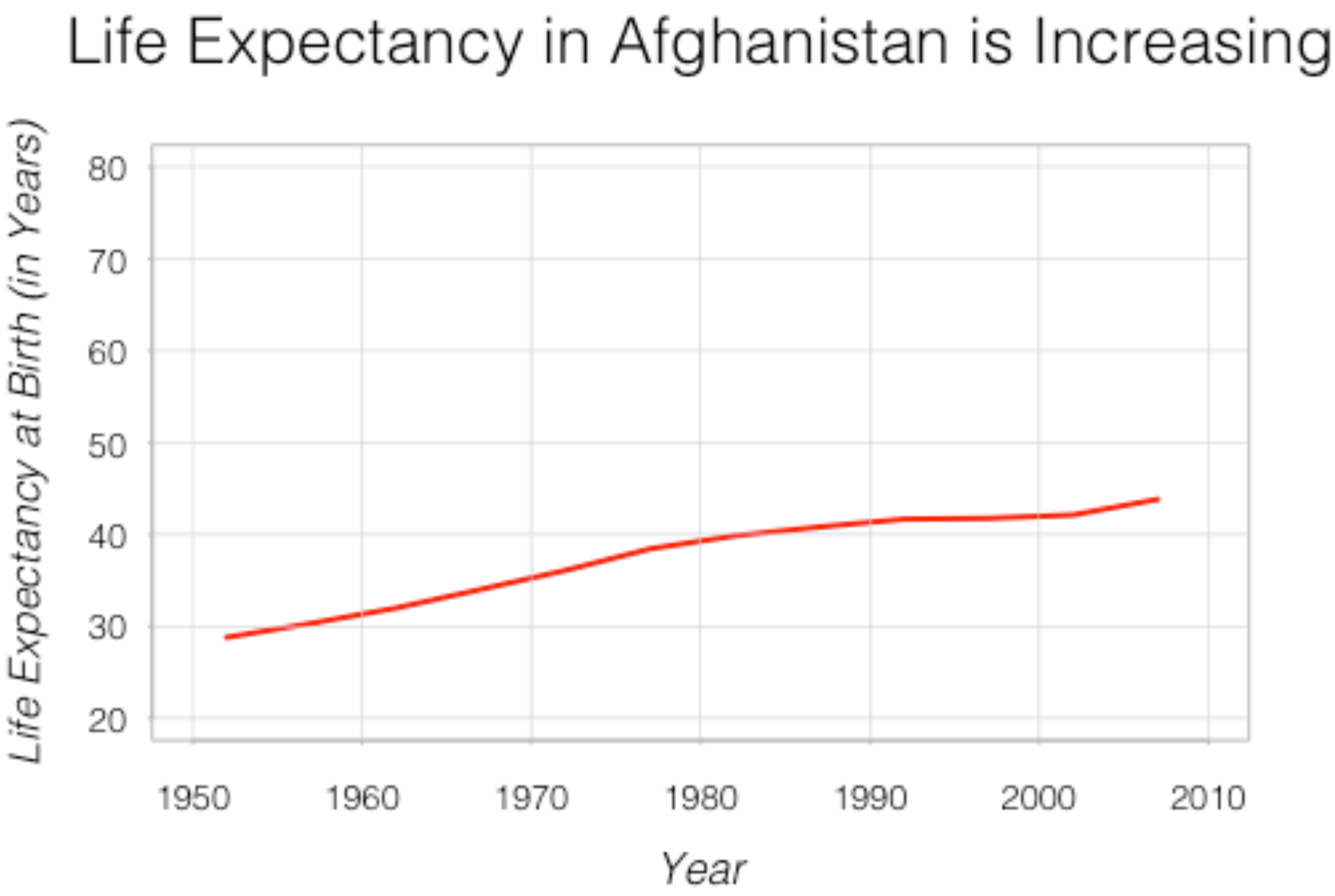
```
par(mgp=c(2, .5, 0), las=1, tck=-0.005, fg="grey")
```



Adding Elements to an Existing Plot

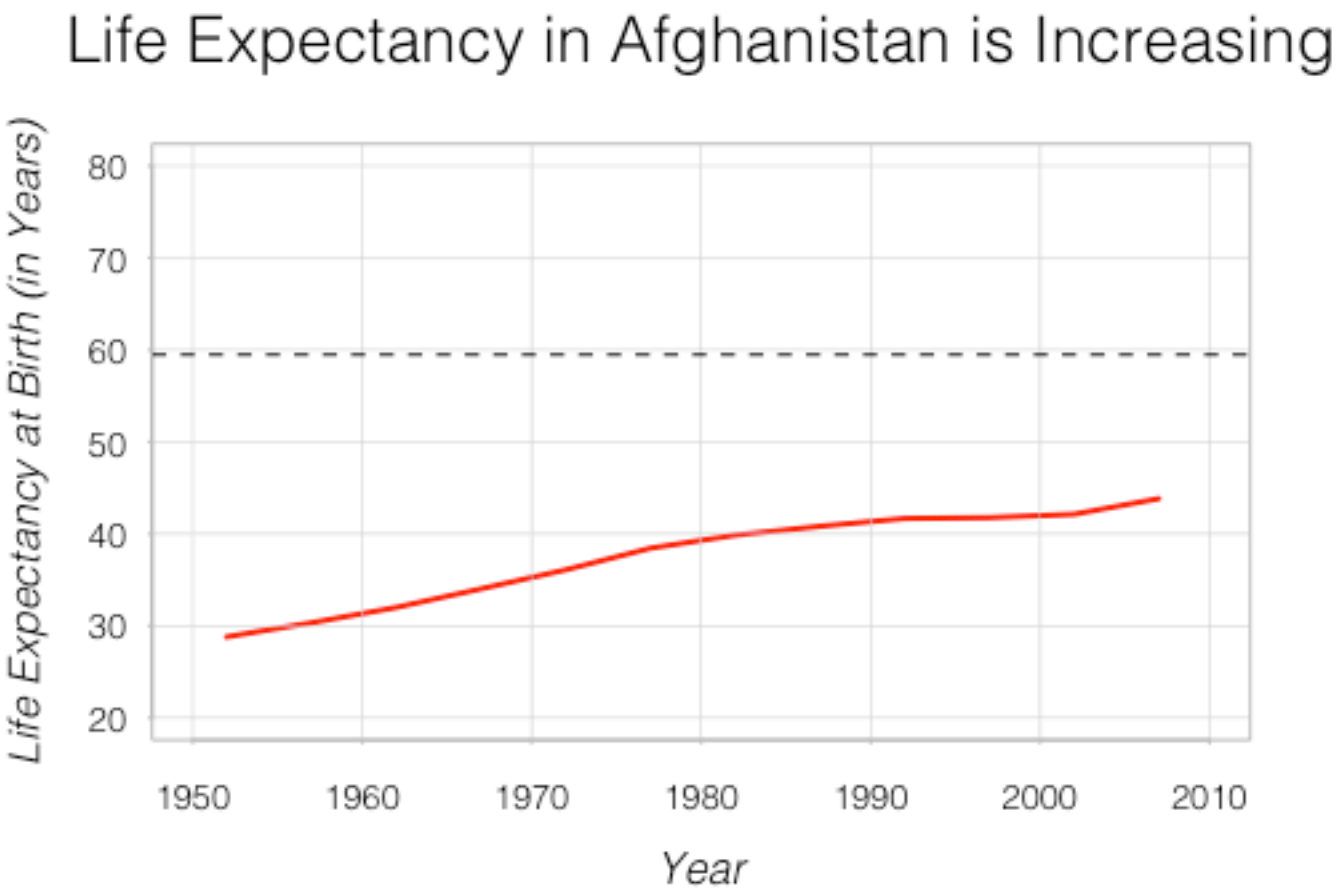
Some Low-level Functions

```
grid(lty=1, lwd=.5)
```



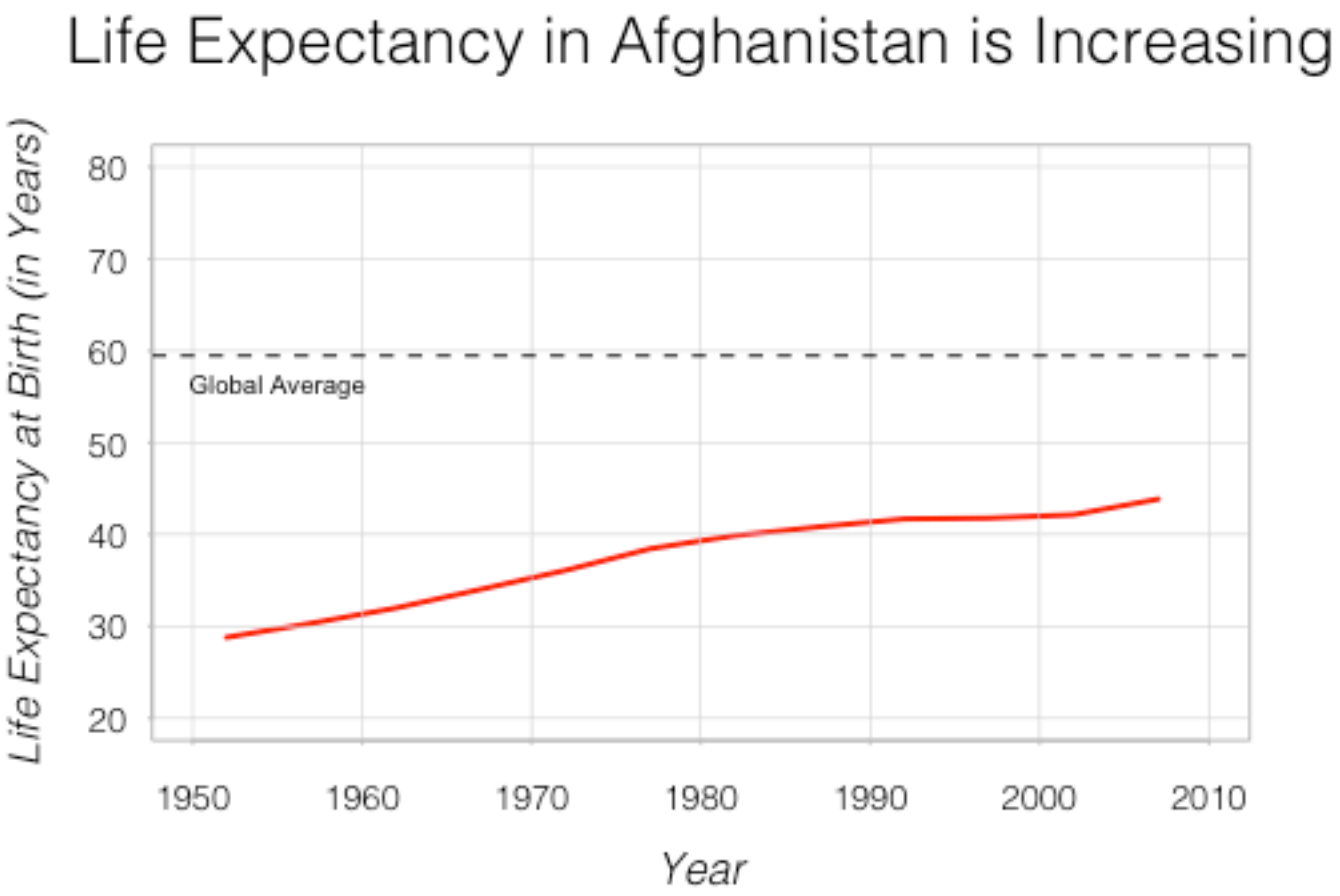
Some Low-level Functions

```
average.life <- mean(data$lifeExp)
abline(h=average.life, lty=2, col="black")
```



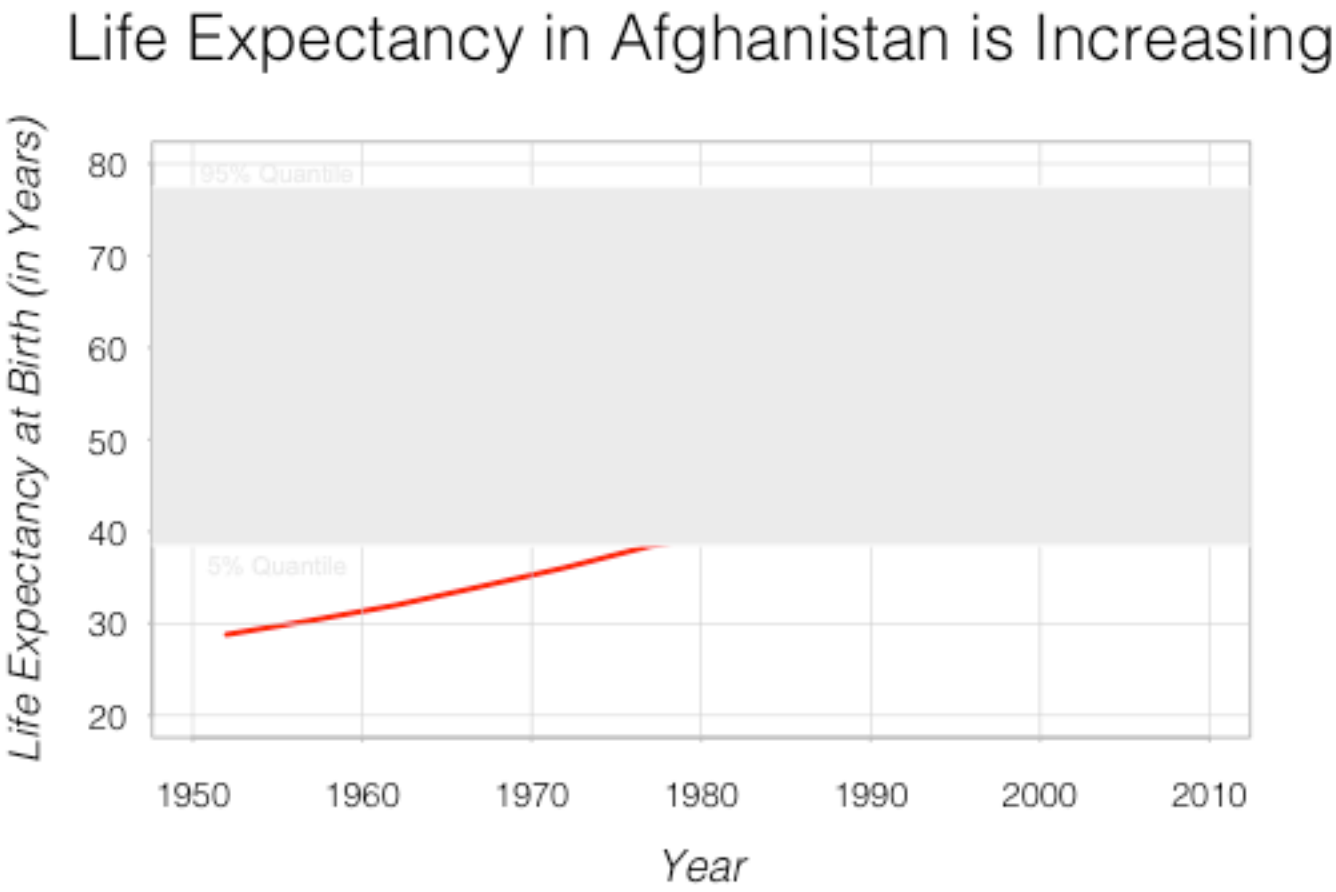
Some Low-level Functions

```
text(1955, 60, "Global Average", col="black", pos=1, cex=.6)
```

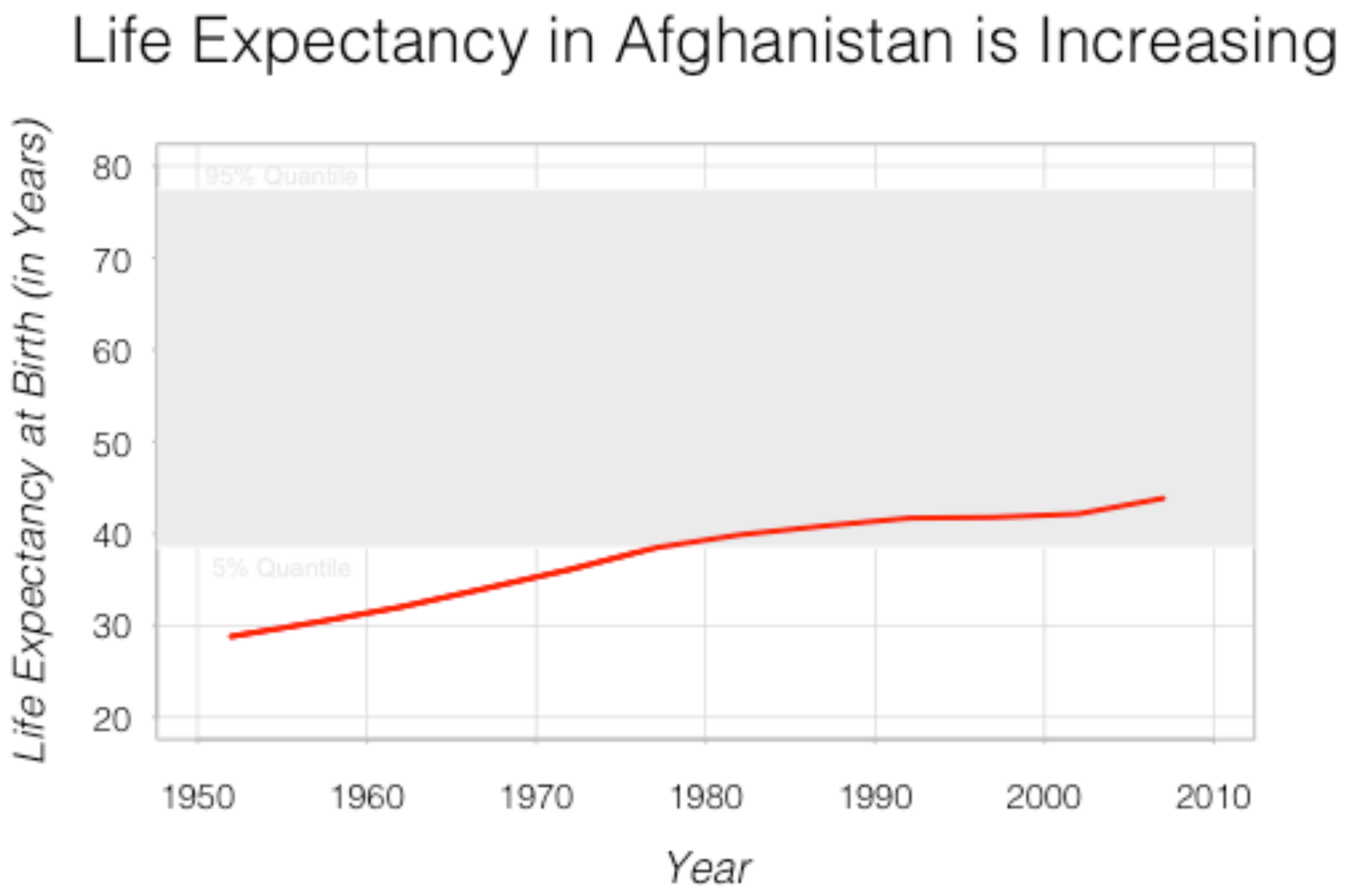


Some Low-level Functions

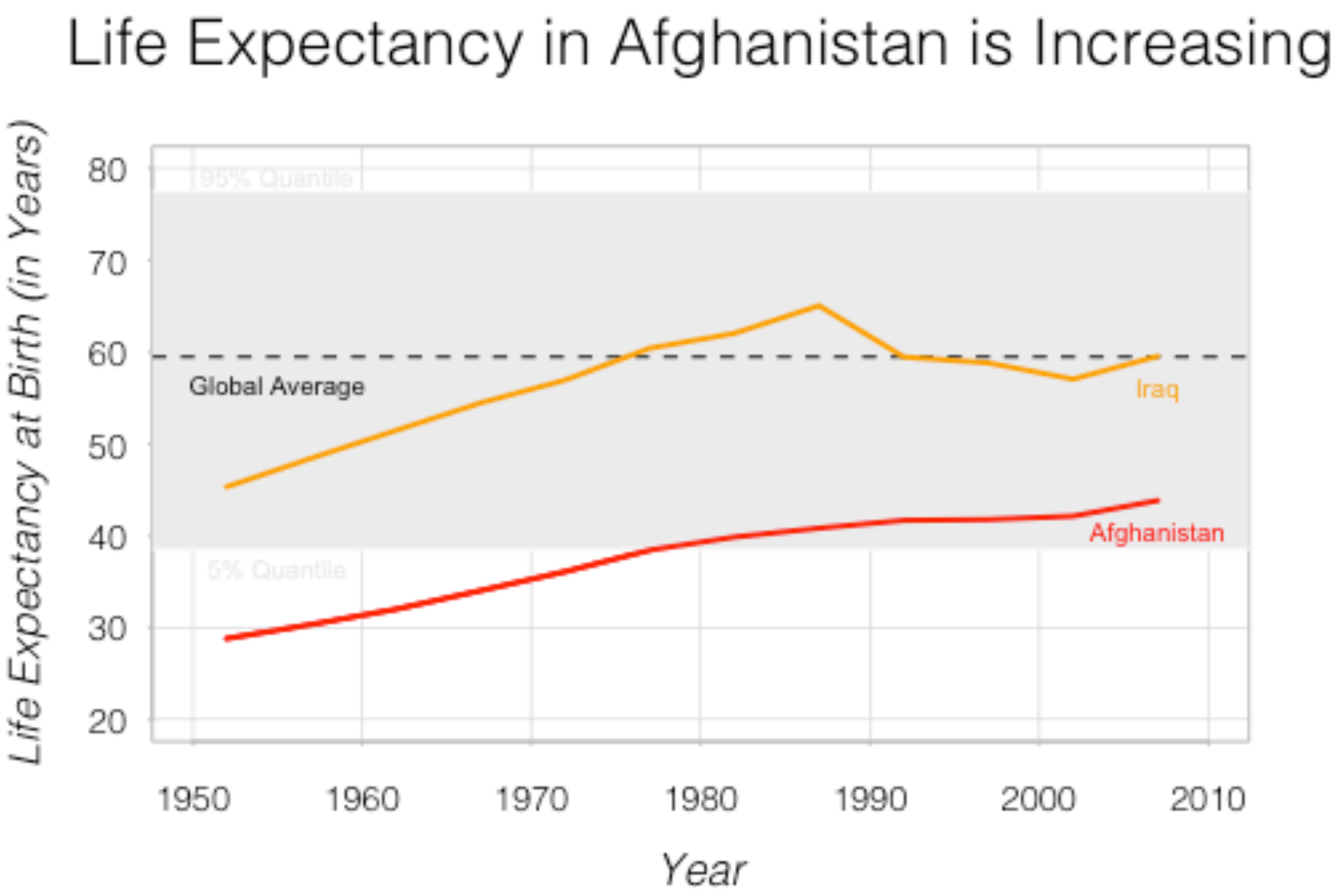
```
normal.life <- quantile(data$lifeExp, c(.05, .95))
rect(1940, normal.life[1], 2020, normal.life[2], col="grey92",
border=F)
```



```
lines(afg$year, afg$lifeExp, lwd=2, col="red")
```



Some Low-level Functions

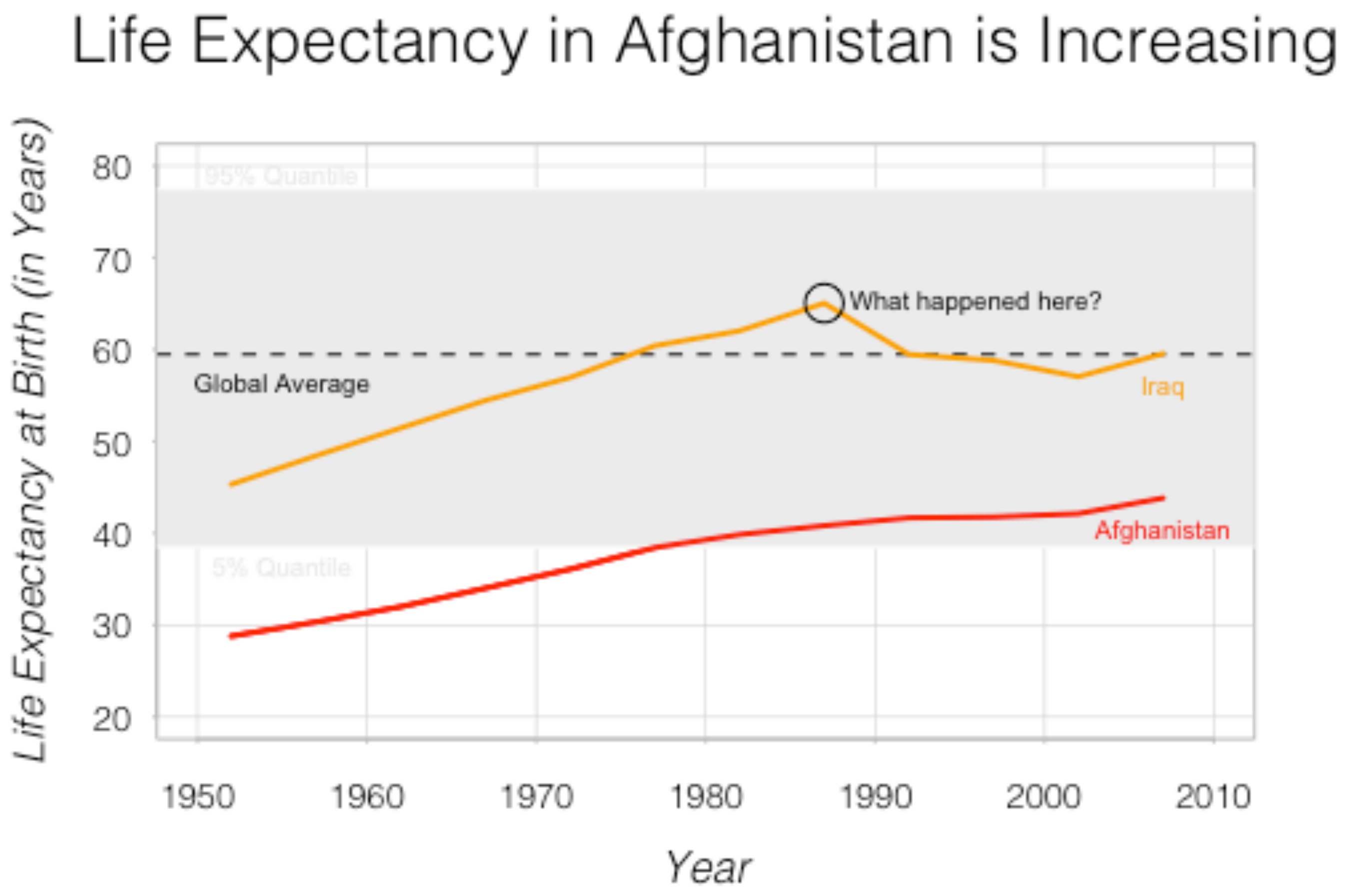


Some Low-level Functions

```
iraq <- data[data$country=="Iraq",]

lines(iraq$year, iraq$lifeExp, lwd=2, col="orange")


text(iraq$year[12], iraq$lifeExp[12], "Iraq", pos=1,
col="orange", cex=.6)

text(afg$year[12], afg$lifeExp[12], "Afghanistan", pos=1,
col="red", cex=.6)
```

Some Low-level Functions

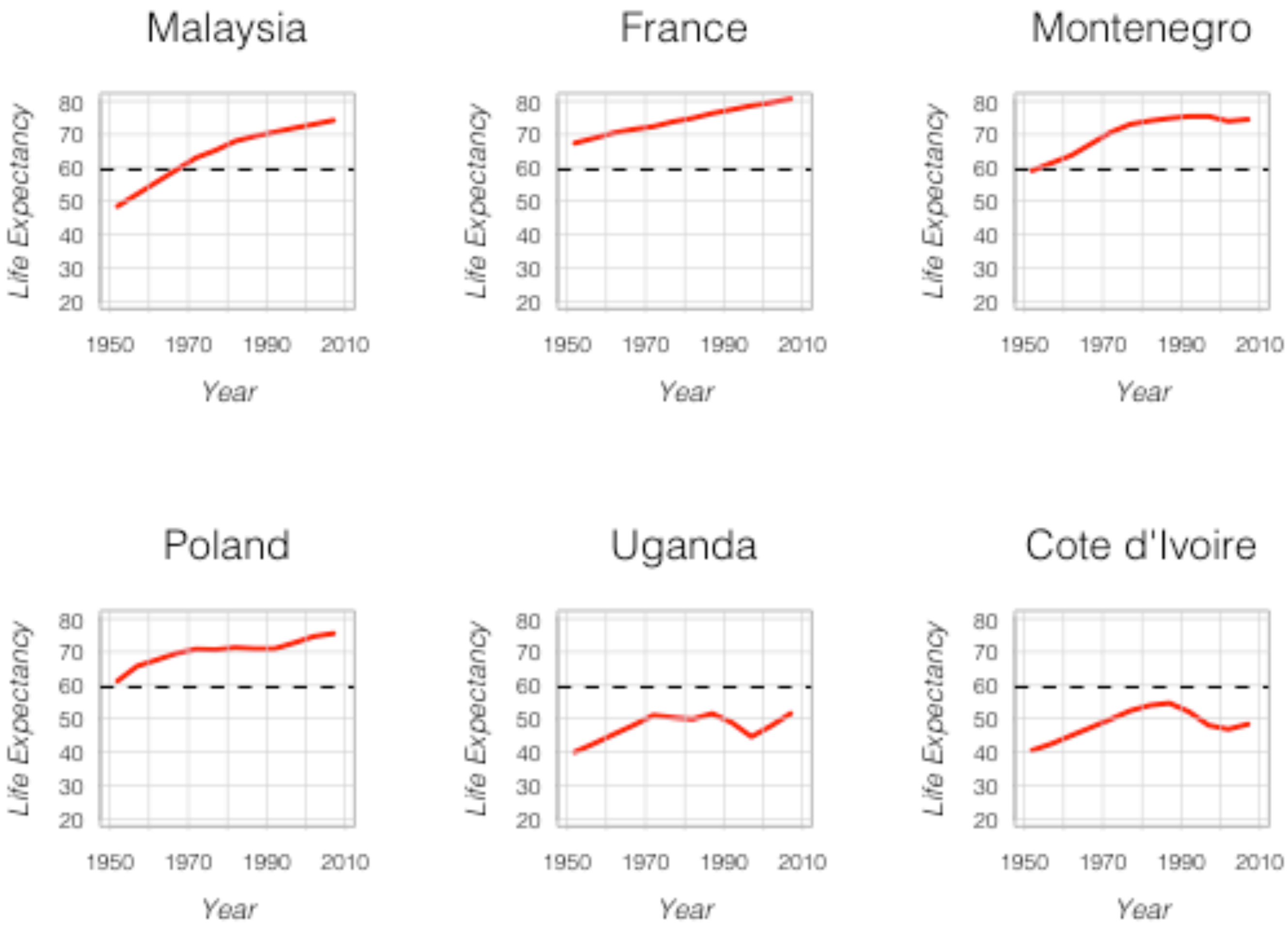
```
points(iraq$year[8], iraq$lifeExp[8], pch=21, cex=2,
col="black")
```



Arranging Multiple Plots

Arranging Multiple Plots

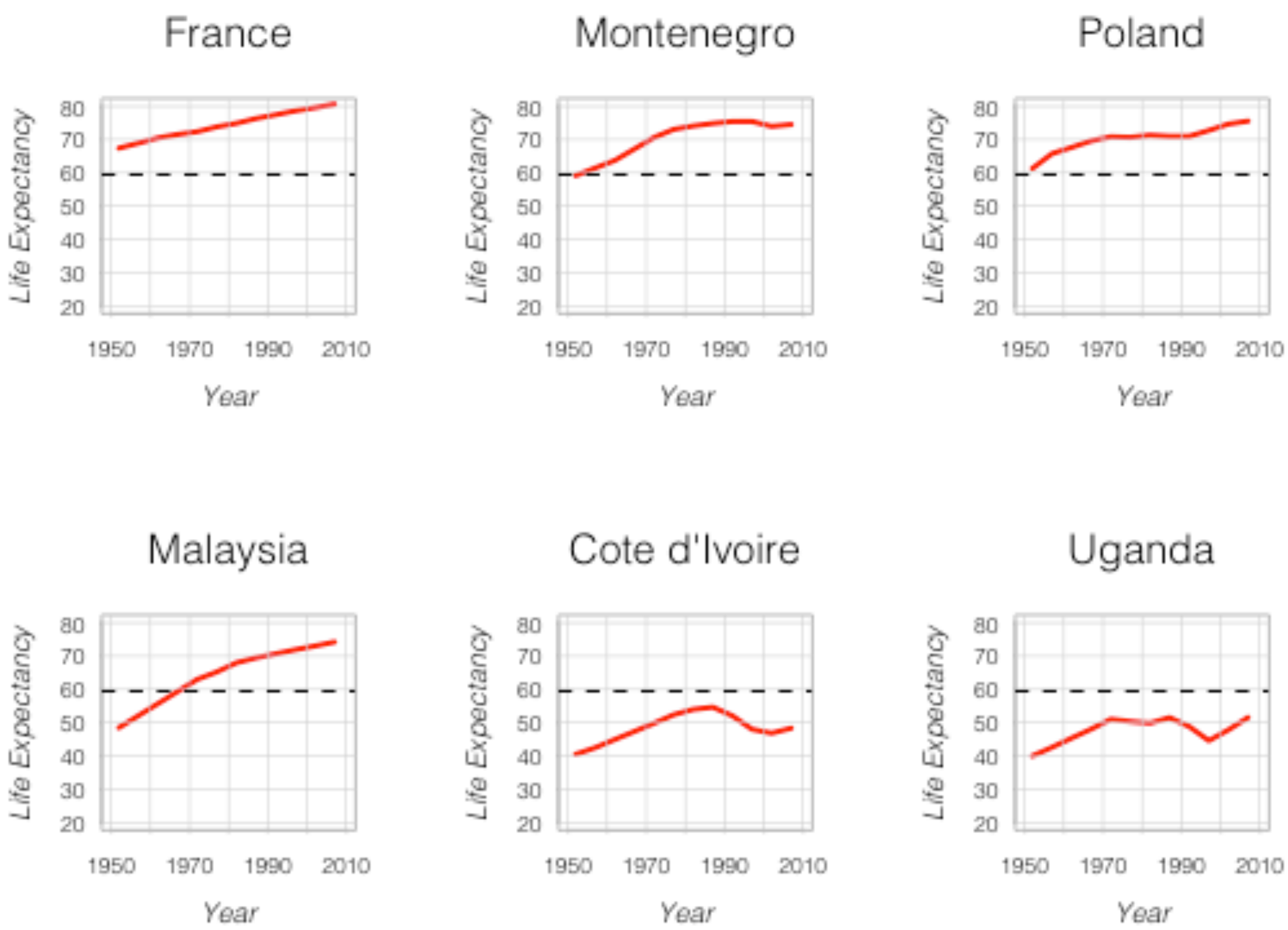
```
par(mfrow=c(2, 3))
```



Arranging Multiple Plots

```
country.samp <- sample(unique(data$country), 6)

for(i in 1:6){

  plot(data$year[data$country==country.samp[i]],
data$lifeExp[data$country==country.samp[i]], type="l", lwd=2, col="red",
     main=country.samp[i],
     xlab="Year", ylab="Life Expectancy",
     ylim=c(20, 80), xlim=c(1950, 2010),
     family="Helvetica Light", font.main=1, font.lab=3,
     cex.main=1.5, cex.lab=1, cex.axis=.8)

  grid(lty=1, lwd=.5)
  abline(h=average.life, lty=2, col="black")
}
```

Arranging Multiple Plots



Arranging Multiple Plots

```
means <- rep(NA, 6)

for(i in 1:6){
  means[i] <- mean(data$lifeExp[data$country==country.samp[i]])
}

ord <- order(means, decreasing=T)

> ord
[1] 2 3 4 1 6

for(i in ord){
  plot(…)
}
```

More in the Lab…