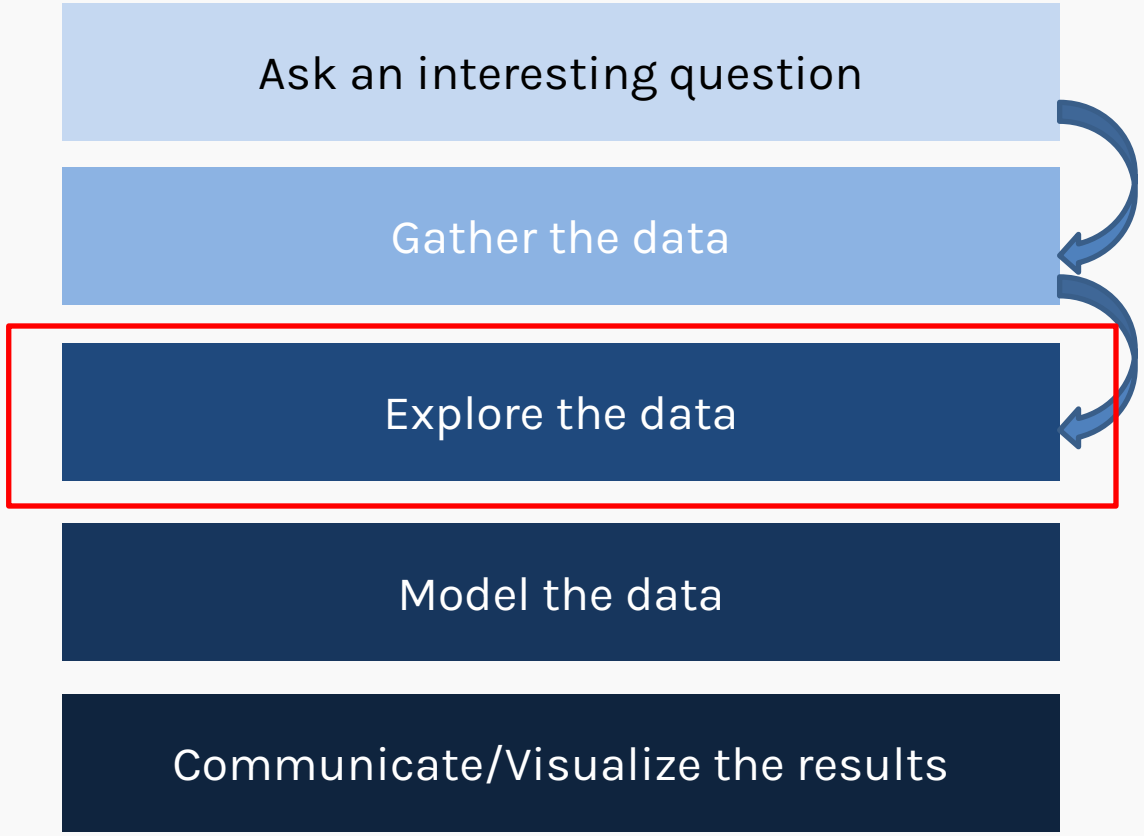
Exploratory Data Analysis

Part A - Effective Exploratory Data Analysis

Pavlos Protopapas

The Data Science Process – Exploring the data



We learnt about cleaning and processing the data, now it is time to explore it.

What is exploratory data analysis (EDA) ?

What is exploratory data analysis (EDA) ?

"It is important to understand what you can do
before you learn to measure how well you seem to have done it."
- *Exploratory Data Analysis(1977)*, John Tukey

EDA techniques include but not limited to understanding summary statistics (mean, median, standard deviations), plotting graphical representations for relation between variables.

Any method of looking at data that does not include formal statistical modeling and inference falls under the term exploratory data analysis.

Why is exploratory data analysis (EDA) important?

Why is exploratory data analysis (EDA) important?

Data is commonly stored in the form of rows and columns and is made up of numbers.

Humans are not very good at gaining insights into the data by looking at huge number of rows of numbers

Nr.	x/c	y/c	Nr.	x/c	y/c	Nr.	x/c	y/c	Nr.	x/c	y/c
1	1.000000	0.001200	24	0.482550	0.084896	47	0.001218	-0.005703	70	0.552264	-0.028223
2	0.998782	0.001477	25	0.447736	0.085686	48	0.004866	-0.010608	71	0.586824	-0.026065
3	0.995134	0.002315	26	0.413176	0.085751	49	0.010926	-0.014852	72	0.620961	-0.023746
4	0.989074	0.003724	27	0.379039	0.085091	50	0.019369	-0.018516	73	0.654509	-0.021328
5	0.980631	0.005714	28	0.345492	0.083847	51	0.030154	-0.021663	74	0.687303	-0.018881
6	0.969846	0.008290	29	0.312697	0.082081	52	0.043227	-0.024350	75	0.719186	-0.016476
7	0.956773	0.011447	30	0.280814	0.079831	53	0.058526	-0.026632	76	0.750000	-0.014185
8	0.941474	0.015175	31	0.250000	0.077136	54	0.075976	-0.028555	77	0.779596	-0.012098
9	0.924024	0.019445	32	0.220404	0.074028	55	0.095492	-0.030165	78	0.807831	-0.010364
10	0.904509	0.024211	33	0.192169	0.070540	56	0.116978	-0.031502	79	0.834565	-0.009048
11	0.883022	0.029413	34	0.165435	0.066698	57	0.140330	-0.032600	80	0.859670	-0.007937
12	0.859670	0.034963	35	0.140330	0.062524	58	0.165435	-0.033488	81	0.883022	-0.006945
13	0.834565	0.040740	36	0.116978	0.058040	59	0.192169	-0.034188	82	0.904509	-0.006035
14	0.807831	0.046572	37	0.095492	0.053257	60	0.220404	-0.034714	83	0.924024	-0.005185
15	0.779596	0.052140	38	0.075976	0.048193	61	0.250000	-0.035074	84	0.941474	-0.004389
16	0.750000	0.057397	39	0.058526	0.042860	62	0.280814	-0.035267	85	0.956773	-0.003651
17	0.719186	0.062374	40	0.043227	0.037270	63	0.312697	-0.035287	86	0.969846	-0.002978
18	0.687303	0.067029	41	0.030154	0.031443	64	0.345492	-0.035123	87	0.980631	-0.002386
19	0.654509	0.071306	42	0.019369	0.025400	65	0.379039	-0.034755	88	0.989074	-0.001892
20	0.620961	0.075142	43	0.010926	0.019172	66	0.413176	-0.034143	89	0.995134	-0.001517
21	0.586824	0.078477	44	0.004866	0.012806	67	0.447736	-0.033160	90	0.998782	-0.001281
22	0.552264	0.081249	45	0.001218	0.006371	68	0.482550	-0.031820	91	1.000000	-0.001200
23	0.517450	0.083403	46	0.000000	0.000000	69	0.517450	-0.030159			

Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.

Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.
- Provide insights into the dataset.

Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.
- Provide insights into the dataset.
- Extract/determine important variables/attributes/features.

Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.
- Provide insights into the dataset.
- Extract/determine important variables/attributes/features.
- Detect outliers and anomalies.

Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.
- Provide insights into the dataset.
- Extract/determine important variables/attributes/features.
- Detect outliers and anomalies.
- Test underlying assumptions.

Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.
- Provide insights into the dataset.
- Extract/determine important variables/attributes/features.
- Detect outliers and anomalies.
- Test underlying assumptions.
- Make informed decisions in developing models.

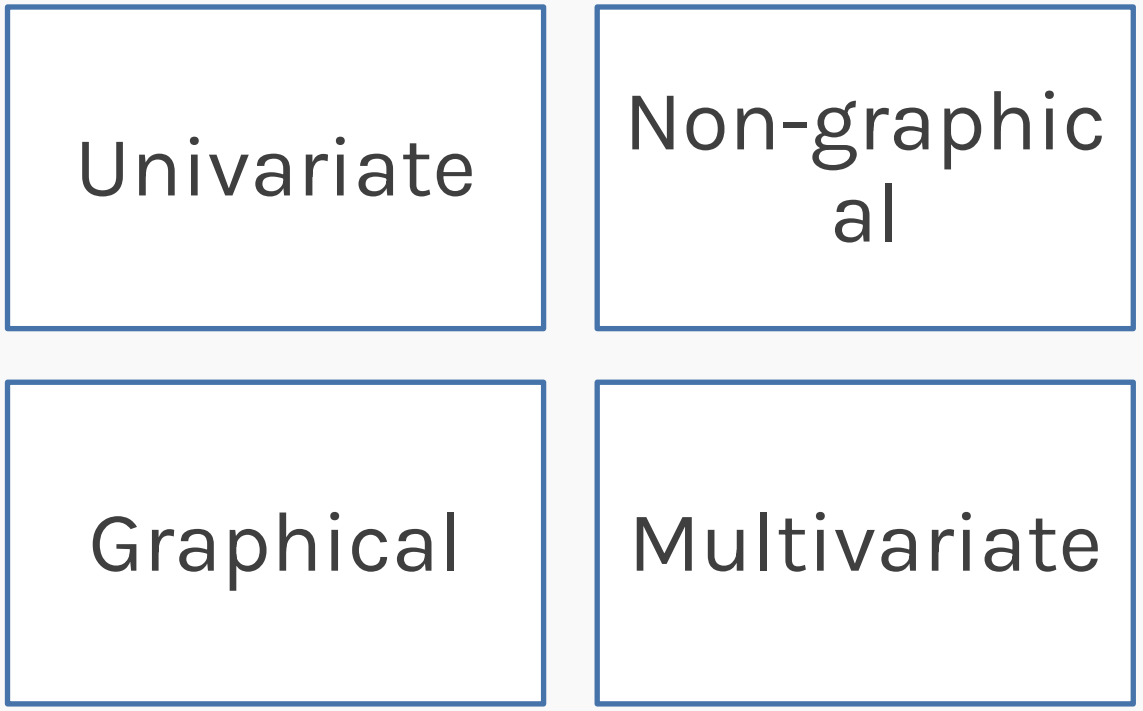
Why is exploratory data analysis (EDA) important?

EDA helps us:

- Ensure our data is expected/valid/appropriate for the task.
- Provide insights into the dataset.
- Extract/determine important variables/attributes/features.
- Detect outliers and anomalies.
- Test underlying assumptions.
- Make informed decisions in developing models.

Types of Exploratory Data Analysis (EDA)

The four types of EDA are univariate non-graphical, multivariate non-graphical, univariate graphical, and multivariate graphical.



Each of the above type can be classified even further based on type of the variables (quantitative or categorical)

Types of Exploratory Data Analysis (EDA)

Let's explore a dataset about cars and it's properties!

Here's a quick look into the dataset

	Height	lenght	width	engine_info	hybrid	forward_gears	fuel_info_city	fuel_type	fuel_info_highway	transmission	make	year	horsepower	torque
0	140	143	202	All-wheel drive	True	6	18	Gasoline	25	Automatic transmission	Audi	2009	250	236
1	140	143	202	Front-wheel drive	True	6	22	Gasoline	28	Automatic transmission	Audi	2009	200	207
2	140	143	202	Front-wheel drive	True	6	21	Gasoline	30	Manual transmission	Audi	2009	200	207
3	140	143	202	All-wheel drive	True	6	21	Gasoline	28	Automatic transmission	Audi	2009	200	207
4	140	143	202	All-wheel drive	True	6	21	Gasoline	28	Automatic transmission	Audi	2009	200	207

Univariate non-graphical EDA

We learn about the characteristics of a single variable like mean, median, mode, standard deviation, variance, etc.

It's always a good idea to inspect each variable individually before doing multivariate EDA

But in order to understand relations between variables we need Multivariate EDA

									torque
count	5076.000000	5076.000000	5076.000000	5076.000000	5076.000000	5076.000000	5076.000000	5076.000000	5076.000000
mean	145.632191	127.825847	144.012411	5.519110	17.275808	24.125493	2010.867612	270.499409	272.707250
std	62.125026	77.358295	79.925899	0.845637	4.479485	6.488293	0.782951	95.293537	100.123328
min	1.000000	2.000000	1.000000	4.000000	8.000000	11.000000	2009.000000	100.000000	98.000000
25%	104.000000	60.000000	62.000000	5.000000	14.000000	20.000000	2010.000000	190.000000	187.000000
50%	152.000000	128.000000	158.000000	6.000000	17.000000	24.000000	2011.000000	266.000000	260.000000
75%	193.000000	198.000000	219.000000	6.000000	20.000000	28.000000	2011.000000	317.000000	335.000000
max	255.000000	255.000000	254.000000	8.000000	38.000000	223.000000	2012.000000	638.000000	774.000000

df.describe()

Multivariate non-graphical EDA

Multivariate non-graphical EDA techniques generally show the relationship between two or more variables in the form of either cross-tabulation or statistics.

We can group our data by one of the categorical variables using one of the aggregate functions(for eg: mean) to understand how categories differ from each other.

fuel_type	Height	lenght	width	forward_gears	horsepower
Compressed natural gas	155.000000	151.000000	216.000000	5.000000	113.000000
Diesel fuel	164.518519	135.407407	171.777778	6.703704	193.148148
E85	123.030702	136.688596	192.377193	5.442982	306.311404
Gasoline	147.761926	126.890873	139.013940	5.519930	267.465912

Cross-tabulation using df.groupby()

Multivariate non-graphical EDA

Correlation matrix

	Height	lenght	width	forward_gears	fuel_info_city	fuel_info_highway	year	horsepower	torque
Height	1.000000	0.051030	-0.220888	0.057769	0.249385	0.245169	0.035920	-0.152407	-0.182470
lenght	0.051030	1.000000	0.016231	-0.041748	-0.018854	-0.010934	-0.031853	0.017107	0.024901
width	-0.220888	0.016231	1.000000	-0.168376	-0.139998	-0.181577	-0.023347	0.036356	0.141035
forward_gears	0.057769	-0.041748	-0.168376	1.000000	-0.036348	0.052588	0.142507	0.319755	0.227386
fuel_info_city	0.249385	-0.018854	-0.139998	-0.036348	1.000000	0.865617	0.092046	-0.701537	-0.754664
fuel_info_highway	0.245169	-0.010934	-0.181577	0.052588	0.865617	1.000000	0.111335	-0.548200	-0.617965
year	0.035920	-0.031853	-0.023347	0.142507	0.092046	0.111335	1.000000	0.006454	-0.019583
horsepower	-0.152407	0.017107	0.036356	0.319755	-0.701537	-0.548200	0.006454	1.000000	0.938304
torque	-0.182470	0.024901	0.141035	0.227386	-0.754664	-0.617965	-0.019583	0.938304	1.000000

A correlation matrix shows correlation coefficients between all possible pairs of variables in the data.

Correlation matrix helps us understand how variables are related to each other.

Why is visualization important?

A DAY IN DATA

The exponential growth of data is undisputed, but the numbers behind this explosion – fuelled by internet of things and the use of connected devcies – are hard to comprehend, particularly when looked at in the context of one day

DEMYSTIFIYING DATA UNITS

From the more familiar 'bit' or 'megabyte', larger units of measurement are more frequently being used to explain the masses of data

Unit		Value	Size
b	bit	0 or 1	1/8 of a byte
B	byte	8 bits	1 byte
KB	kilobyte	1,000 bytes	1,000 bytes
MB	megabyte	$1,000^2$ bytes	1,000,000 bytes
GB	gigabyte	$1,000^3$ bytes	1,000,000,000 bytes
TB	terabyte	$1,000^4$ bytes	1,000,000,000,000 bytes
PB	petabyte	$1,000^5$ bytes	1,000,000,000,000,000 bytes
EB	exabyte	$1,000^6$ bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	$1,000^7$ bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	$1,000^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

*A lowercase "b" is used as an abbreviation for bits, while an uppercase "B" represents bytes.

463EB

of data will be created every day by 2025

IDC

500m

tweets are sent every day

Twitter

4PB

of data created by Facebook, including

350m photos

100m hours of video watch time

Facebook Research

95m

photos and videos are shared on Instagram

Instagram Business

320bn

emails to be sent each day by 2021

294bn

billion emails are sent

306bn

emails to be sent each day by 2020

Radicati Group

65bn

messages sent over WhatsApp and two billion minutes of voice and video calls made

Facebook

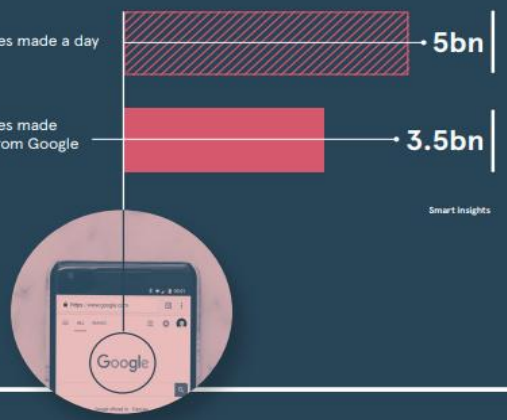
3.9bn

people use emails

4TB

of data produced by a connected car

Intel



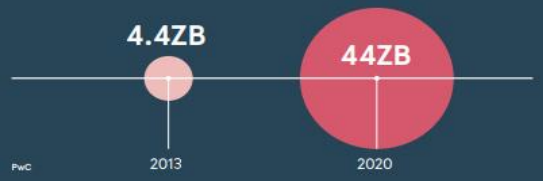
Smart Insights

28PB

to be generated from wearable devices by 2020

Statista

ACCUMULATED DIGITAL UNIVERSE OF DATA



PwC

Credits: World Economic Forum

RACONTEUR

What is all that data?

Enormous amounts of data are produced and stored at various ends from:

- Physical sensors.
- Health and medicinal records.
- Records of human activity.
- Financial transactions, etc

What is all that data?

Enormous amounts of data are produced and stored at various ends from:

- Physical sensors.
- Health and medicinal records.
- Records of human activity.
- Financial transactions, etc

Imagine solving data science problems and analyzing large amounts of data!! 🤯

Visualization to the rescue 📊📉📈

We might we use visualization to empower understanding of data and analysis processes.

Visualization facilitates the discovery of questions and answers to them simultaneously.

Visualization to the rescue 📊📉📈

We might we use visualization to empower understanding of data and analysis processes.

Visualization facilitates the discovery of questions and answers to them simultaneously.

Furthermore, we may use visualization for:

- Recording information.
- Making decisions.
- Discover patterns.
- Convey information.

Visualization to the rescue 📊📉📈

We might we use visualization to empower understanding of data and analysis processes.

Visualization facilitates the discovery of questions and answers to them simultaneously.

Well, these are just words. We'll witness the wonders of visualization ourselves. Just wait and watch!

Fu

- Recording information.
- Making decisions.
- Discover patterns.
- Convey information.

EDA and visualization aiding human understanding

- Humans can't help but look for patterns and find structure in the information coming their way.

- Visualizations are the single easiest way for our brains to receive and interpret large amounts of information.

Our brains are natural pattern recognition machines! The human brain has evolved to recognize patterns, perhaps more than any other single function. Our brain is weak at processing logic, remembering facts, and making calculations, but pattern recognition is its deep core capability.

Human brain as pattern detection machine

Human brain as pattern detection machine

Oh! It looks like a submarine!

Human brain as pattern detection machine

Oh! It looks like a submarine!

Or maybe a whale?

Human brain as pattern detection machine

Oh! It looks like a submarine!

Umm… just random clouds?

Or maybe a whale?

Human brain as pattern detection machine

Human brain as pattern detection machine

Human brain as pattern detection machine

Human brain as pattern detection machine

Analysis Example: Antibiotics Effectiveness
Will Burtin, 1951

In the fall of 1951, Will Burtin published data showing the effectiveness of three popular antibiotics on 16 different bacteria, measured in terms of minimum inhibitory concentration.

Data: Collected prior 1951

	Antibiotic			
Bacteria	**Penicillin**	**Streptomycin**	**Neomycin**	**Gram Staining**
Aerobacter aerogenes	870	1	1.6	negative
Brucella abortus	1	2	0.02	negative
Brucella antracis	0.001	0.01	0.007	positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae	850	1.2	1	negative
Mycobacterium tuberculosis	800	5	2	negative
Proteus vulgaris	3	0.1	0.1	negative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus	0.007	0.1	0.001	positive
Staphylococcus aureus	0.03	0.03	0.001	positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

Data: Collected prior 1951

Genus of bacteria

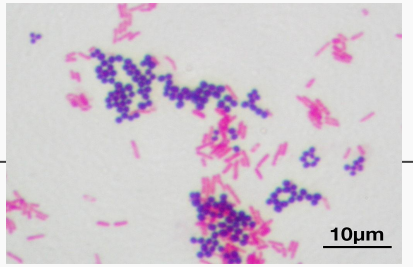
Bacteria
Aerobacter aerogenes
Brucella abortus
Brucella antracis
Diplococcus pneumoniae
Escherichia coli
Klebsiella pneumoniae
Mycobacterium tuberculosis
Proteus vulgaris
Pseudomonas aeruginosa
Salmonella (Eberthella) typhosa
Salmonella schottmuelleri
Staphylococcus albus
Staphylococcus aureus
Streptococcus fecalis
Streptococcus hemolyticus
Streptococcus viridans

Data: Collected prior 1951

Antibiotic applied

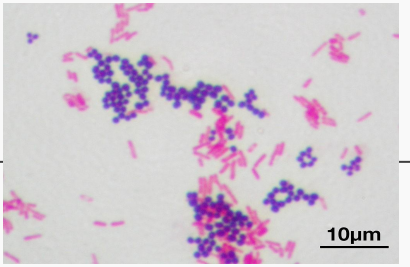
	Antibiotic		
Bacteria	Penicillin	Streptomycin	Neomycin
Aerobacter aerogenes	870	1	1.6
Brucella abortus	1	2	0.02
Brucella antracis	0.001	0.01	0.007
Diplococcus pneumoniae	0.005	11	10
Escherichia coli	100	0.4	0.1
Klebsiella pneumoniae	850	1.2	1
Mycobacterium tuberculosis	800	5	2
Proteus vulgaris	3	0.1	0.1
Pseudomonas aeruginosa	850	2	0.4
Salmonella (Eberthella) typhosa	1	0.4	0.008
Salmonella schottmuelleri	10	0.8	0.09
Staphylococcus albus	0.007	0.1	0.001
Staphylococcus aureus	0.03	0.03	0.001
Streptococcus fecalis	1	1	0.1
Streptococcus hemolyticus	0.001	14	10
Streptococcus viridans	0.005	10	40

Data: Collected prior 1951



	Antibiotic			
Bacteria	Penicillin	Streptomycin	Neomycin	Gram Staining
Aerobacter aerogenes	870	1	1.6	negative
Brucella abortus	1	2	0.02	negative
Brucella antracis	0.001	0.01	0.007	positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae	850	1.2	1	negative
Mycobacterium tuberculosis	800	5	2	negative
Proteus vulgaris	3	0.1	0.1	negative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus	0.007	0.1	0.001	positive
Staphylococcus aureus	0.03	0.03	0.001	positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

Data: Collected prior 1951



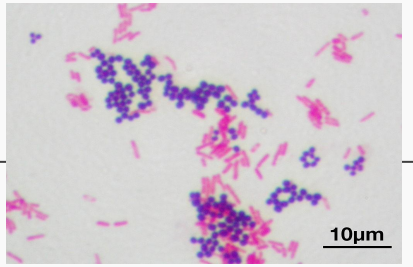
Genus of bacteria

Antibiotic applied

Bacteria	Antibiotic: Penicillin	Streptomycin	Neomycin	Gram Staining
Aerobacter aerogenes	870	1	1.6	negative
Brucella abortus	1	2	0.02	negative
Brucella antracis	0.001	0.01	0.007	positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae	850	1.2	1	negative
Mycobacterium tuberculosis	800	5	2	negative
Proteus vulgaris	3	0.1	0.1	negative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus	0.007	0.1	0.001	positive
Staphylococcus aureus	0.03	0.03	0.001	positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

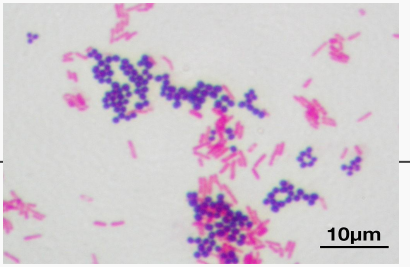
Min. Inhibitory Concentration [ml/g]

What questions might we ask?



	Antibiotic			
Bacteria	Penicillin	Streptomycin	Neomycin	Gram Staining
Aerobacter aerogenes	870	1	1.6	negative
Brucella abortus	1	2	0.02	negative
Brucella antracis	0.001	0.01	0.007	positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae	850	1.2	1	negative
Mycobacterium tuberculosis	800	5	2	negative
Proteus vulgaris	3	0.1	0.1	negative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus	0.007	0.1	0.001	positive
Staphylococcus aureus	0.03	0.03	0.001	positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

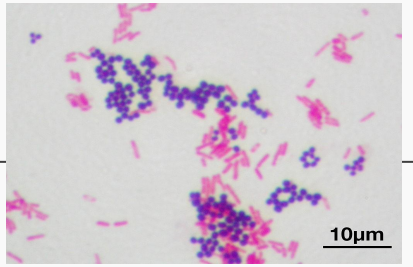
What questions might we ask?



Bacteria	Antibiotic			Gram Staining
	Penicillin	Streptomycin	Neomycin	
Aerobacter aerogenes				negative
Brucella abortus				negative
Brucella antracis				positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae	850	1.2	1	negative
Mycobacterium tuberculosis	800	5	2	negative
Proteus vulgaris	3	0.1	0.1	negative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus	0.007	0.1	0.001	positive
Staphylococcus aureus	0.03	0.03	0.001	Positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

How do the drugs compare?

What questions might we ask?

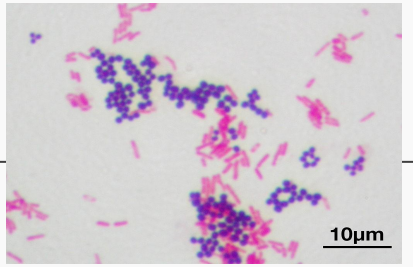


Bacteria	Antibiotic			Gram Staining
	Penicillin	Streptomycin	Neomycin	
Aerobacter aerogenes				negative
Brucella abortus				negative
Brucella antracis				positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae	850	1.2	1	negative
Mycobacteri				egative
Proteus vulg				egative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus	0.007	0.1	0.001	positive
Staphylococcus aureus	0.03	0.03	0.001	Positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

How do the drugs compare?

Do the bacteria group by antibiotic resistance?

What questions might we ask?



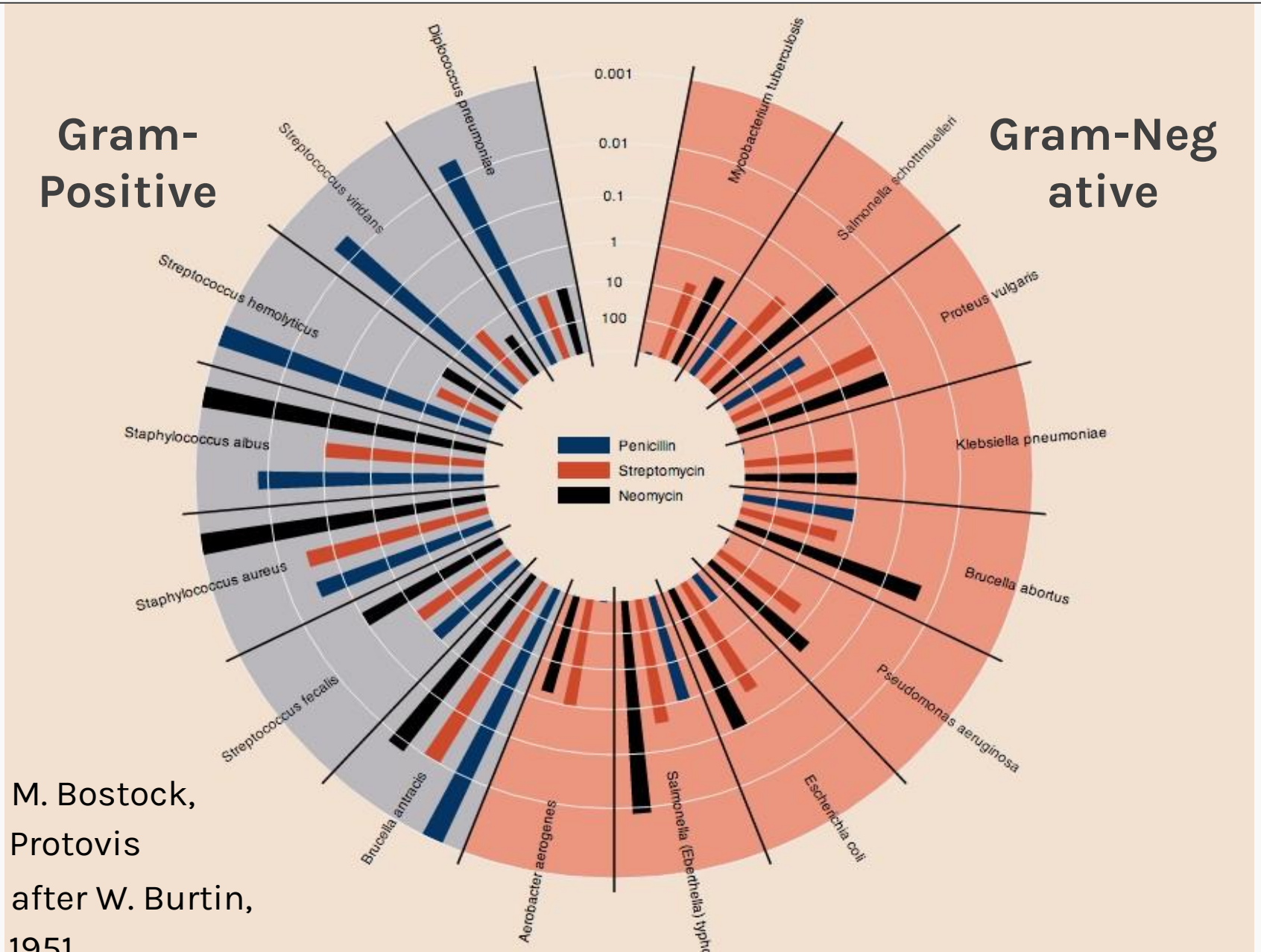
Bacteria	Antibiotic			Gram Staining
	Penicillin	Streptomycin	Neomycin	
Aerobacter aerogenes				negative
Brucella abortus				negative
Brucella antracis				positive
Diplococcus pneumoniae	0.005	11	10	positive
Escherichia coli	100	0.4	0.1	negative
Klebsiella pneumoniae				negative
Mycobacteri				egative
Proteus vulg				egative
Pseudomonas aeruginosa	850	2	0.4	negative
Salmonella (Eberthella) typhosa	1	0.4	0.008	negative
Salmonella schottmuelleri	10	0.8	0.09	negative
Staphylococcus albus				positive
Staphylococcus aureu				Positive
Streptococcus fecalis	1	1	0.1	positive
Streptococcus hemolyticus	0.001	14	10	positive
Streptococcus viridans	0.005	10	40	positive

How do the drugs compare?

Do the bacteria group by antibiotic resistance?

Which drug should one use?

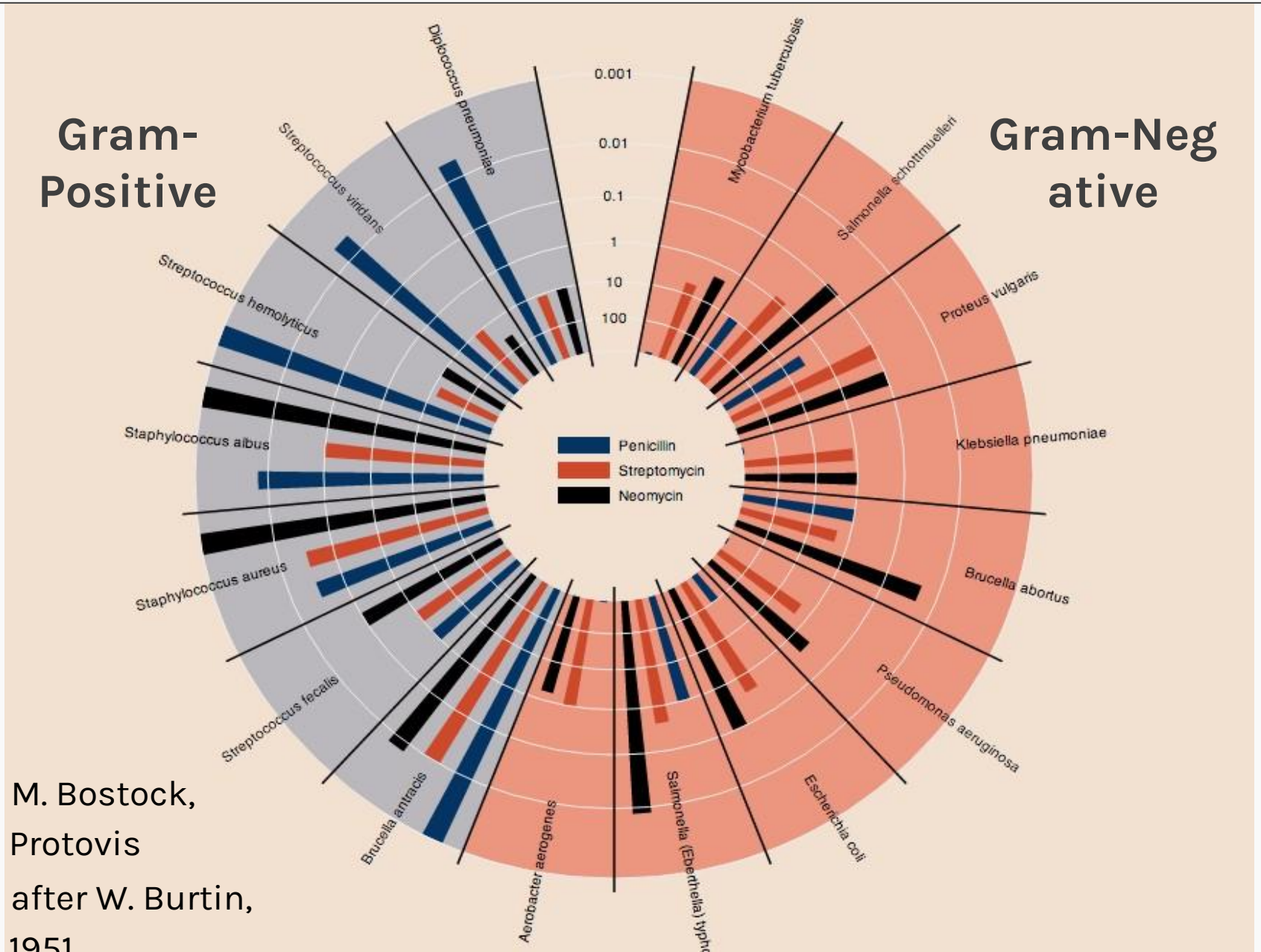
How do the drugs compare?



M. Bostock, Protovis after W. Burtin, 1951

- Each sector contains the bacteria's inhibition to each antibiotic.
- Gram-positive bacteria has been positioned on the left and Gram-negative bacteria has been positioned on the right.

How do the drugs compare?



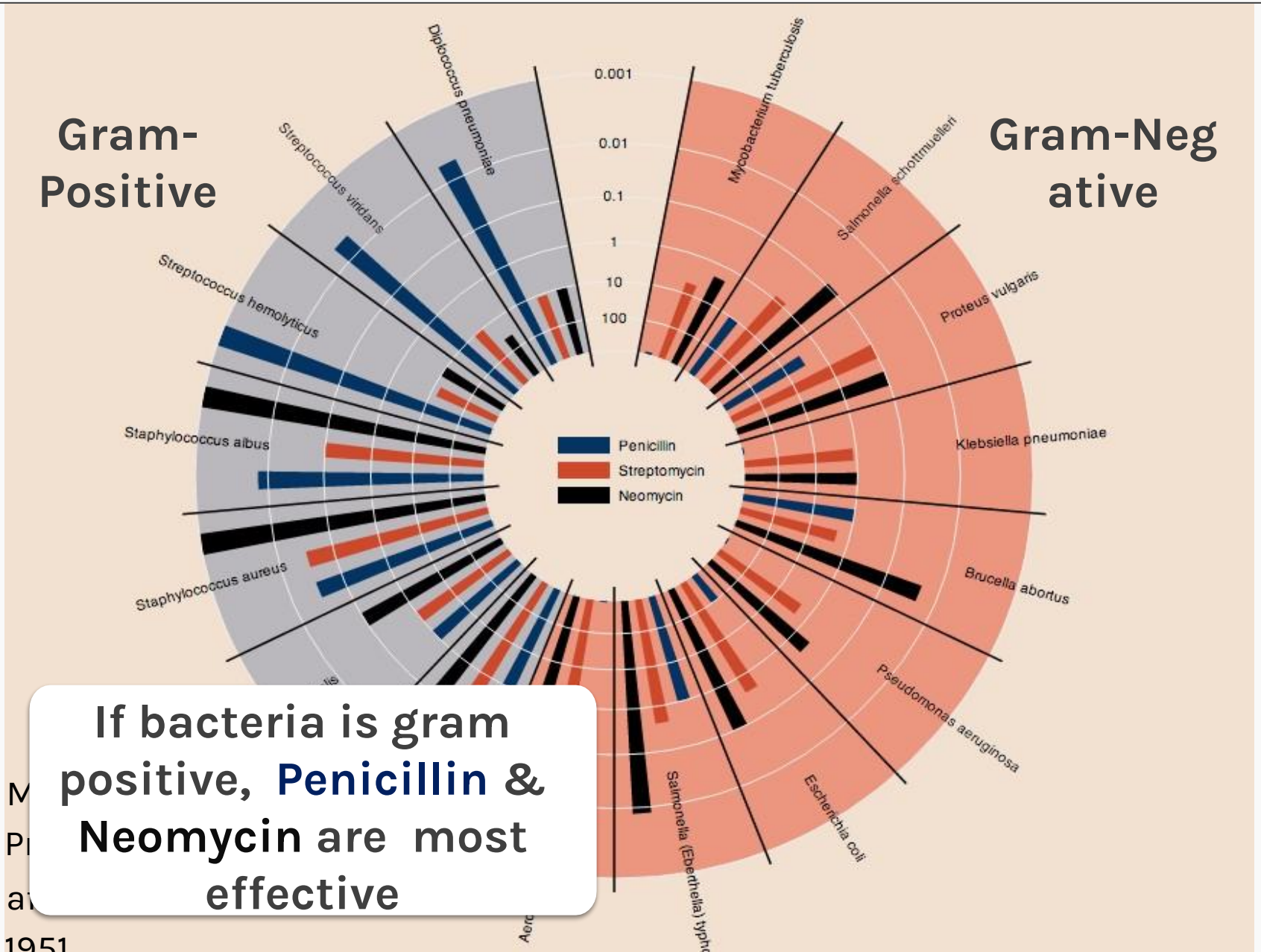
M. Bostock, Protovis

after W. Burtin, 1951

Encodings

Radius: 1 / log(MIC)
Bar Color: Antibiotic
Background Color: Gram Staining
Bar length: proportional to effectiveness of a drug.

How do the drugs compare?

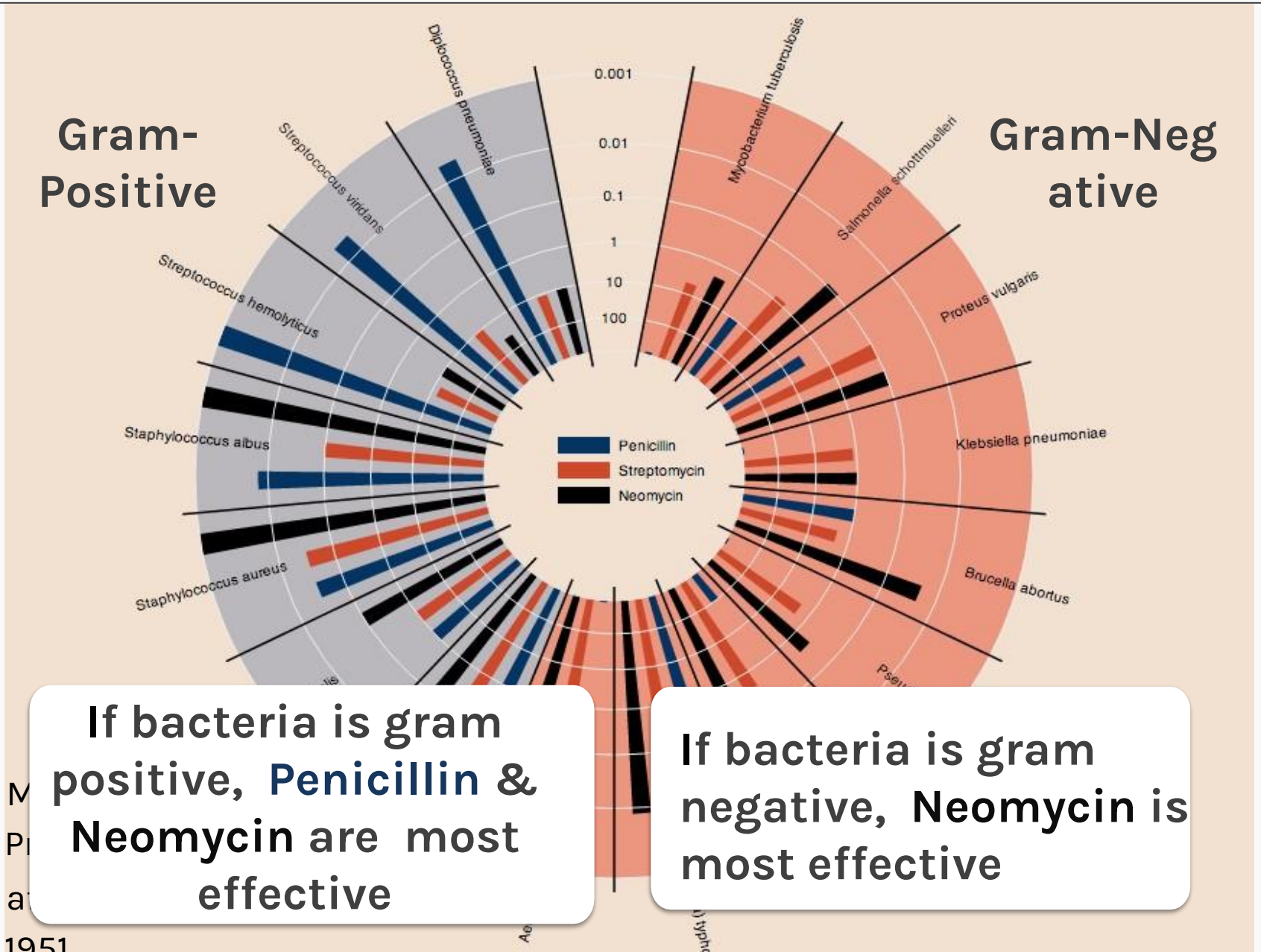


If bacteria is gram positive, Penicillin & Neomycin are most effective

Encodings

Radius: 1 / log(MIC)
Bar Color: Antibiotic
Background Color: Gram Staining
Bar length: proportional to effectiveness of a drug.

How do the drugs compare?



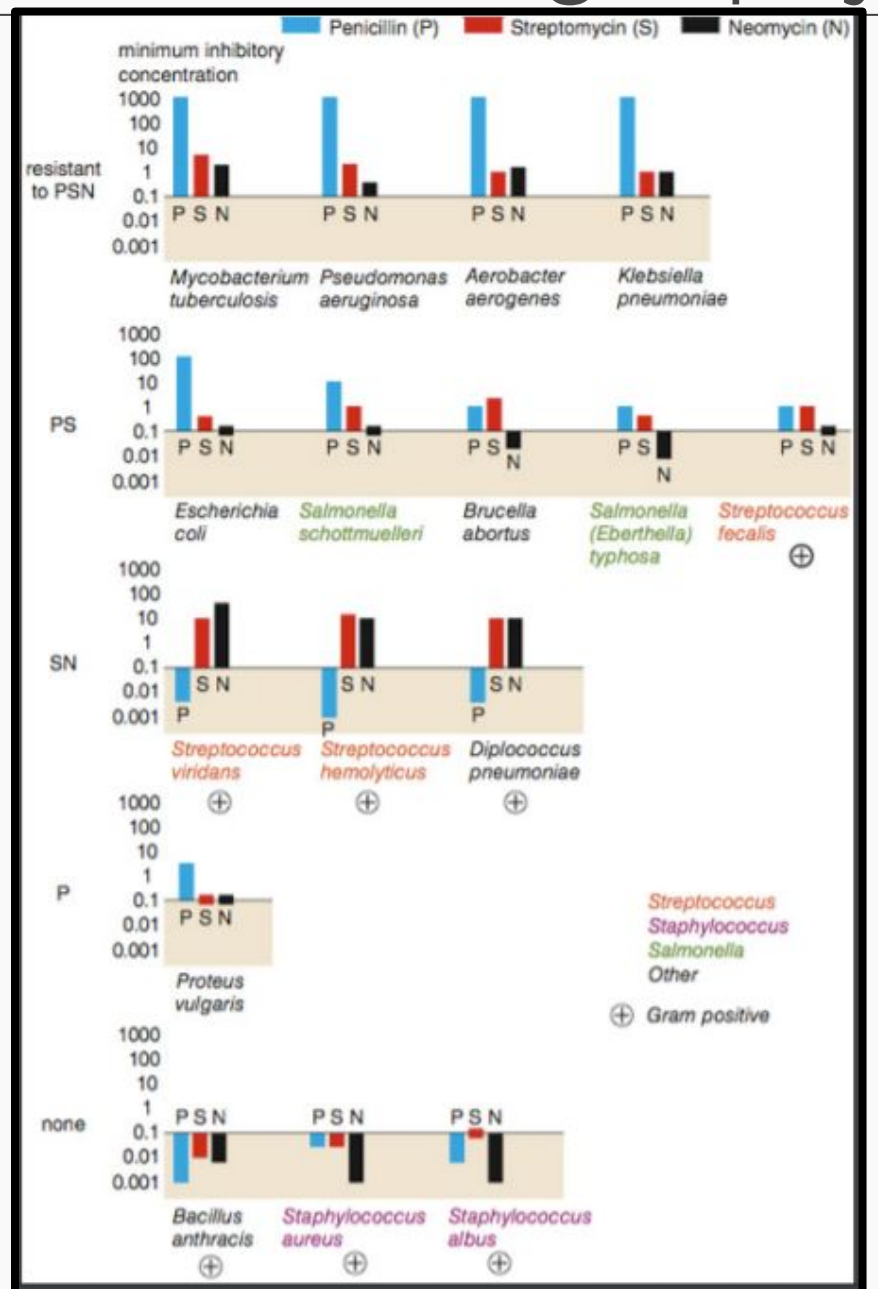
If bacteria is gram positive, Penicillin & Neomycin are most effective

If bacteria is gram negative, Neomycin is most effective

Encodings

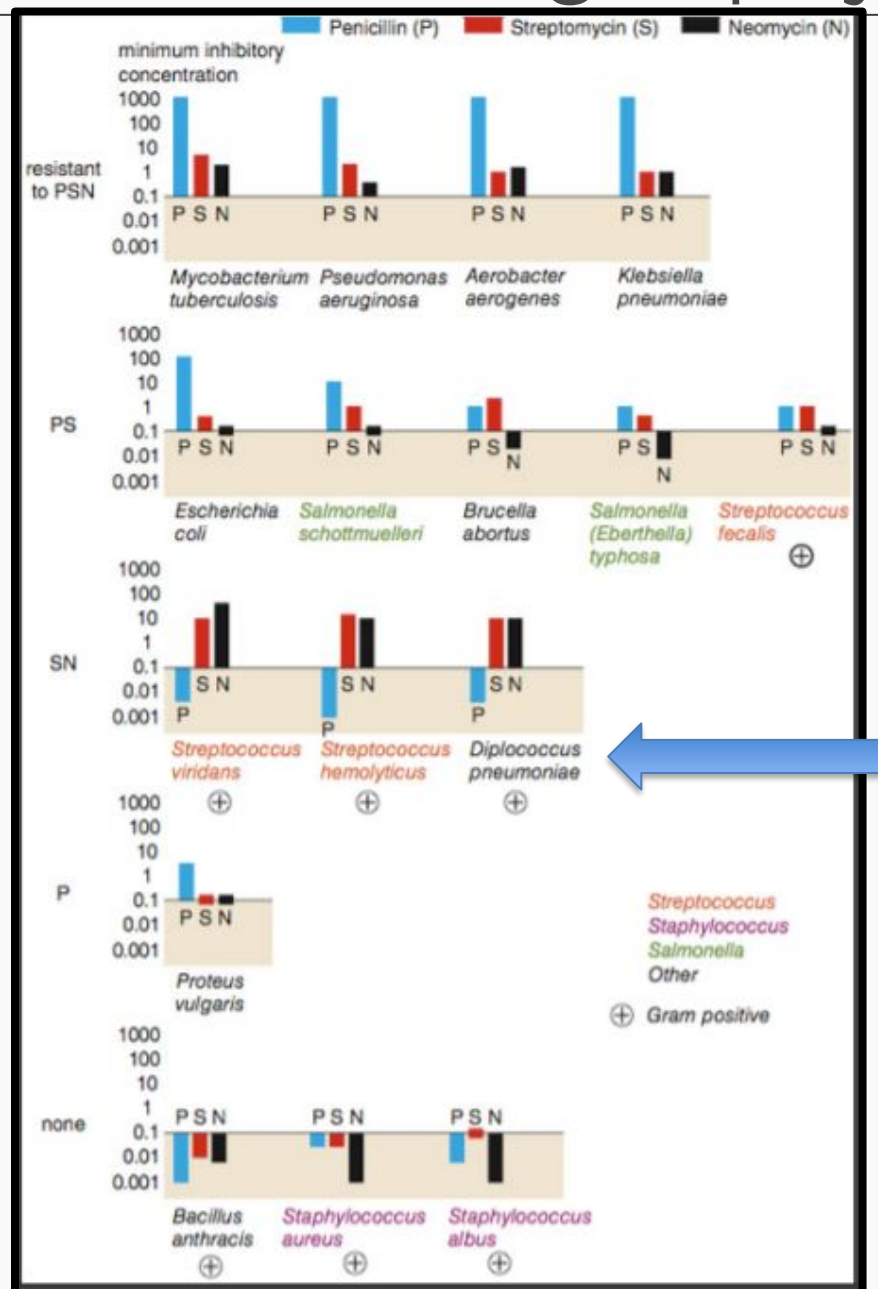
Radius: 1 / log(MIC)
Bar Color: Antibiotic
Background Color: Gram Staining
Bar length: proportional to effectiveness of a drug.

Do the bacteria group by antibiotic resistance?



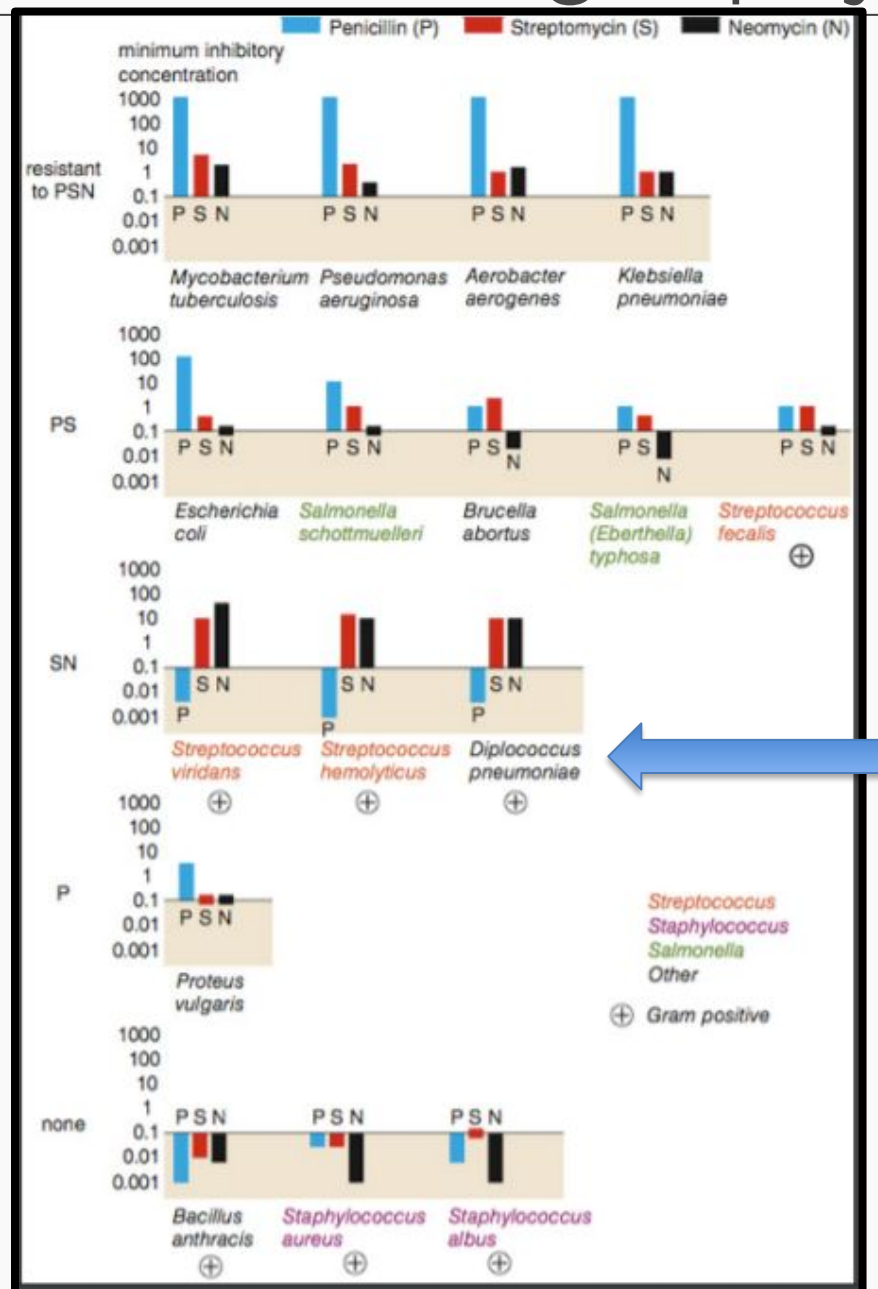
- Note that the y-axis is now inverted
- Bacteria are grouped by their resistance to the three antibiotics

Do the bacteria group by antibiotic resistance?



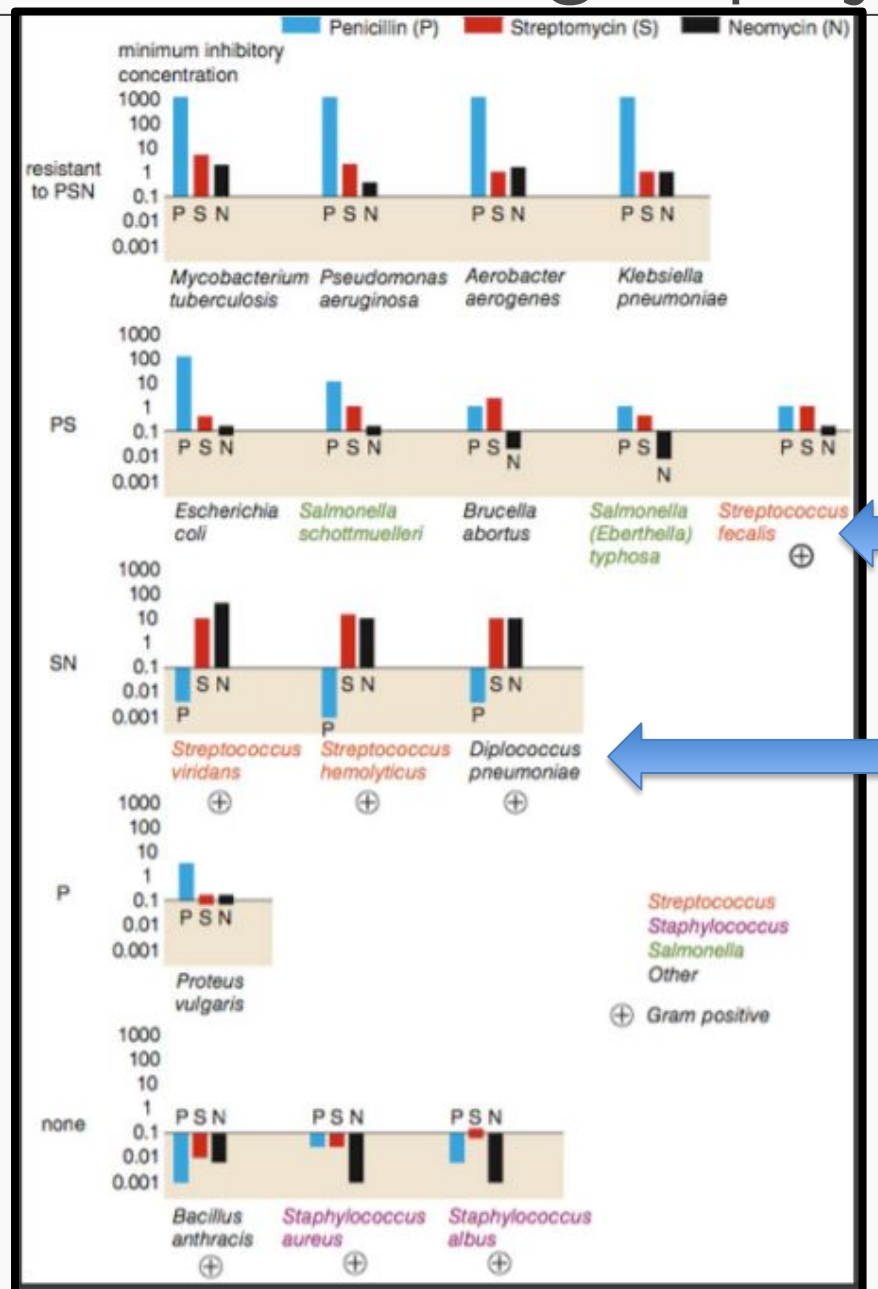
- Note that the y-axis is now inverted
- Bacteria are grouped by their resistance to the three antibiotics

Do the bacteria group by antibiotic resistance?



Really a streptococcus!
(realized ~20 years later)

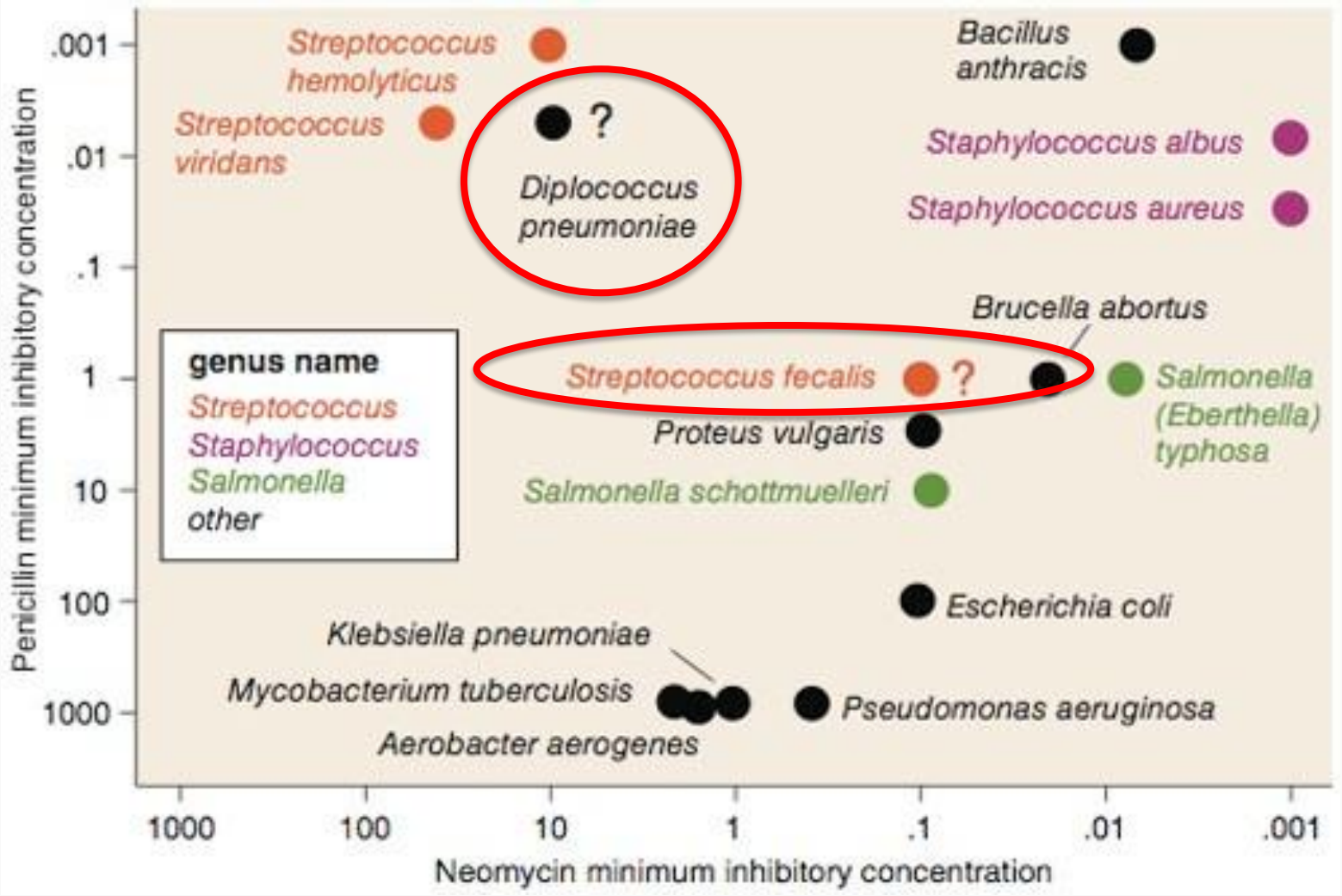
Do the bacteria group by antibiotic resistance?



Not a streptococcus!
(realized ~30 years later)

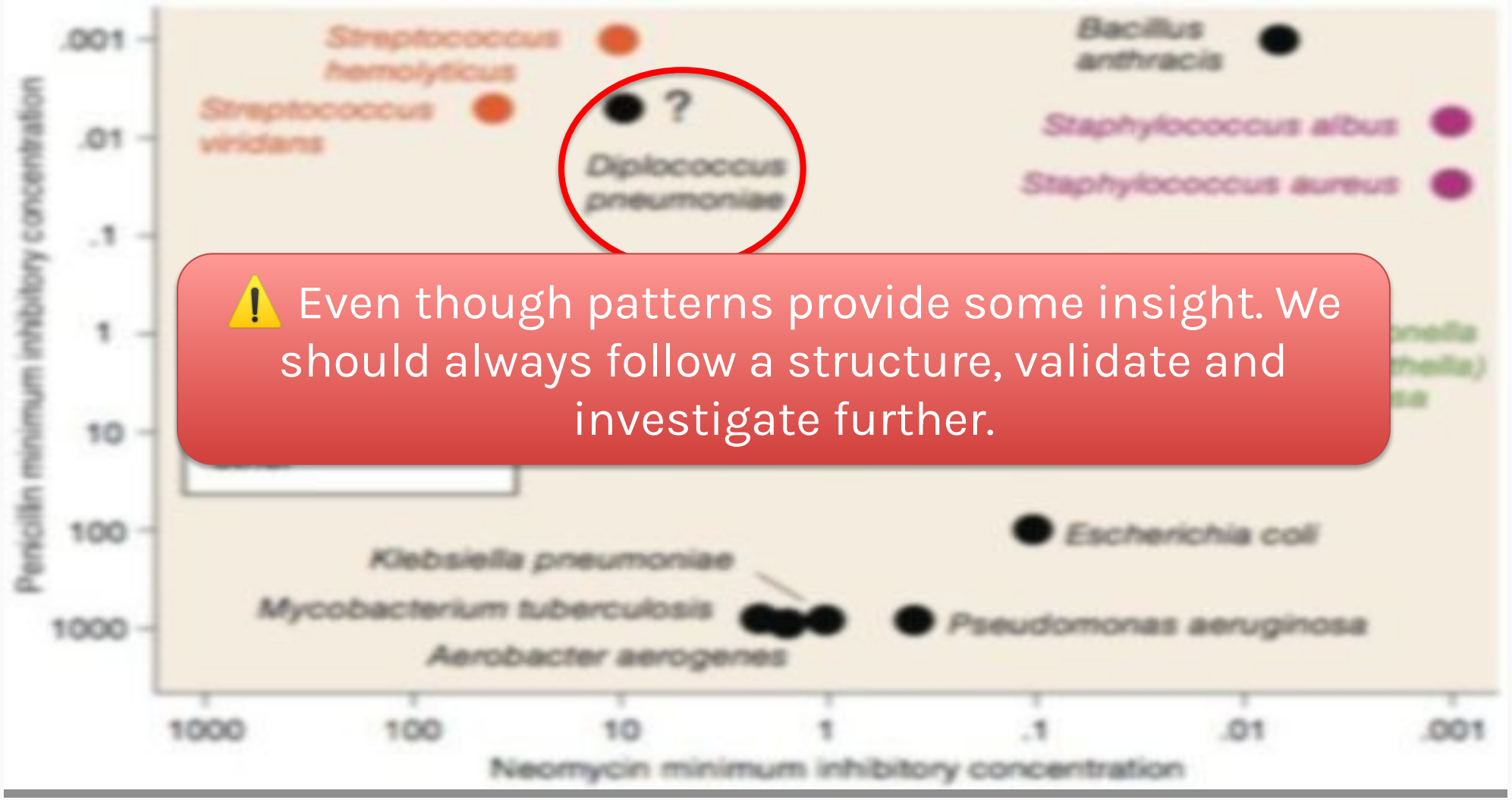
Really a streptococcus!
(realized ~20 years later)

How do the drugs compare?



Wainer & Lysen, "That's funny…" American

How do the drugs compare?



⚠️ Even though patterns provide some insight. We should always follow a structure, validate and investigate further.

Wainer & Lysen, “That’s funny…” American

Let's consolidate everything we've learnt so far and give it a structure

Exploratory Data Analysis (EDA) Workflow

EDA Workflow

1. **Build** the dataframe from the data. (Ideally, include all the data)

EDA Workflow

1. **Build** the dataframe from the data. (Ideally, include all the data)
2. **Clean** the dataframe. It should have the following properties:
    - Each row describes a single object.
    - Each column describes a property of the object.
    - Columns are numeric whenever appropriate.
    - Columns contain atomic properties that cannot be further decomposed.

EDA Workflow (contd.)

3. Explore **global properties.** Use histograms, scatter plots, and aggregation functions to summarize the data.

EDA Workflow (contd.)

3. Explore **global properties.** Use histograms, scatter plots, and aggregation functions to summarize the data.
4. Explore **group properties.** Use groupby and small multiples to compare subsets of data.

EDA Example: The English Premier League

Credits: Harvard’s CS109A

EDA Example: The English Premier League

For instance, we are interested in the English Premier League (football/soccer) and we want to build a model to predict a player's market value.

The question is: Does age affect one's market value?

Build the dataframe

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31		22

- **Credible/Trustworthy?**
- **Possibly subjective market values?**
- **Sampled data**

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent		31	CB	22

Does it contain the necessary information?

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

Missing data? Imputation needed.

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

Are the data types okay (df.dtypes)? Should be casted?

from www.transfermarkt.us

Explore the data

name	club	age	position	market value
Alexis Sanchez	Arsenal	28	LW	65
Mesut Ozil	Arsenal	28	AM	50
Petr Cech	Arsenal	35	GK	7
Theo Walcott	Arsenal	28	RW	20
Laurent Koscielny	Arsenal	31	CB	22

Are the values reasonable? `DataFrame.describe()` ...

from www.transfermarkt.us

Explore the data

	age	page_views	fpl_value	fpl_points	market_value
count	461.000000	461.000000	461.000000	461.000000	461.000000
mean	26.804772	763.776573	5.447939	57.314534	11.012039
std	3.961892	931.805757	1.346695	53.113811	12.257403
min	17.000000	3.000000	4.000000	0.000000	0.050000
25%	24.000000	220.000000	4.500000	5.000000	3.000000
50%	27.000000	460.000000	5.000000	51.000000	7.000000
75%	30.000000	896.000000	5.500000	94.000000	15.000000
max	38.000000	7664.000000	12.500000	264.000000	75.000000

Are the values reasonable? `DataFrame.describe()` **...**

Explore the data

	age	page_views	fpl_value	fpl_points	market_value
count	461.00000				
mean	26.80477				
std	3.961892	931.805757	1.346695	53.113811	2.257403
min	17.000000	3.000000	4.000000	0.000000	0.050000
25%	24.000000	220.000000	4.500000	5.000000	3.000000
50%	27.000000	460.000000	5.000000	51.000000	7.000000
75%	30.000000	896.000000	5.500000	94.000000	15.000000
max	38.000000	7664.000000	12.500000	264.000000	75.000000

This seems abnormally low. Is it correct? Who is this?

Are the values reasonable? `DataFrame.describe()` **...**

Explore the data

	age	page_views	fpl_value	fpl_points	market_value
co					461.000000
m					11.012039
std					12.257403
min	17.000000	3.000000	4.000000	0.000000	0.050000
25%	24.000000	220.000000	4.500000	5.000000	3.000000
50%	27.000000	460.000000	5.000000	51.000000	7.000000
75%	30.000000	896.000000	5.500000	94.000000	15.000000
max	38.000000	7664.000000	12.500000	264.000000	75.000000

This also seems suspicious. Is it correct? Who is this?

Are the values reasonable? `DataFrame.describe()` ...

Inspecting suspicious data

This accounts for both extreme values that we noticed. But, is this data **truly accurate?** It's worth validating online, elsewhere.

```
import pandas as pd
df = pd.read_csv("epl.csv")
df.iloc[df['market_value'].idxmin()]
```

```
name                Eduardo Carvalho
club                         Chelsea
age                             34
position                        LW
position_cat                     1
market_value                  0.05
page_views                     467
fpl_value                        5
fpl_sel                      0.10%
fpl_points                       0
region                           2
nationality                Portugal
new_foreign                      0
age_cat                          6
club_id                          5
big_club                         1
new_signing                      1
Name: 109, dtype: object
```

Inspecting suspicious data

	age	page_views	fpl_value	fpl_points	market_value
count	461.000000	461.000000	461.000000	461.000000	461.000000
mean					
std					
min					
25%	24.000000	220.000000	4.500000	5.000000	3.000000
50%	27.000000	460.000000	5.000000	51.000000	7.000000
75%	30.000000	896.000000	5.500000	94.000000	15.000000
max	38.000000	7664.000000	12.500000	264.000000	75.000000

What is going on here?! Is someone actually paid that much more than others? And someone scores that much more?

Are the values reasonable? `DataFrame.describe()` ...

Explore the data

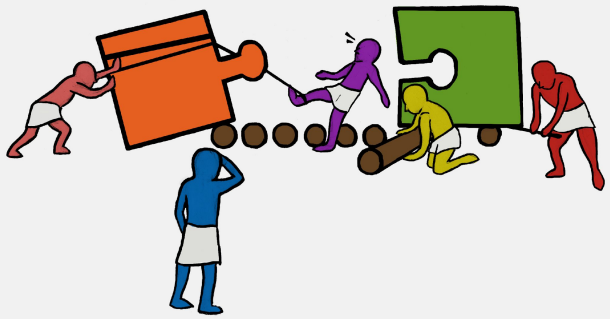
```
df.loc[df['market_value'] >= 15].sort_values(by='market_value', ascending=False).head(15)
```

	name	club	age	position	position_cat	market_value	age_views	fpl_value	fpl_se	fpl_points
92	Eden Hazard	Chelsea	26	LW	1	75.0	4220	10.5	2.30%	224
263	Paul Pogba	Manchester+United	24	CM	2	75.0	7435	8.0	19.50%	115
0	Alexis Sanchez	Arsenal	28	LW	1	65.0	4329	12.0	17.10%	264
241	Sergio Aguero	Manchester+City	29	CF	1	65.0	4046	11.5	9.70%	175
240	Kevin De Bruyne	Manchester+City	26	AM	1	65.0	2252	10.0	17.50%	199
377	Harry Kane	Tottenham	23	CF	1	60.0	4161	12.5	35.10%	224
104	N%27Golo Kante	Chelsea	26	DM	2	50.0	4042	5.0	13.80%	83
1	Mesut Ozil	Arsenal	28	AM	1	50.0	4395	9.5	5.60%	167
260	Romelu Lukaku	Manchester+United	24	CF	1	50.0	3727	11.5	45.00%	221
93	Diego Costa	Chelsea	28	CF	1	50.0	4454	10.0	3.00%	196
214	Philippe Coutinho	Liverpool	25	AM	1	45.0	2958	9.0	30.80%	171
242	Raheem Sterling	Manchester+City	22	LW	1	45.0	2074	8.0	3.80%	149
376	Dele Alli	Tottenham	21	CM	2	45.0	4626	9.5	38.60%	225
98	Thibaut Courtois	Chelsea	25	GK	4	40.0	1260	5.5	18.50%	141
215	Sadio Mane	Liverpool	25	LW	1	40.0	3219	9.5	5.30%	156

Explore the data

	age	page_views	fpl_value	fpl_points	market_value
count	461.000000	461.000000	461.000000	461.000000	461.000000
mean	26.804772	763.776573	5.447939	57.314534	11.012039
std	3.961892	931.805757	1.346695	53.113811	12.257403
min	17.000000	3.000000	4.000000	0.000000	0.050000
25%	24.000000	220.000000	4.500000	5.000000	3.000000
50%	27.000000	460.000000	5.000000	51.000000	7.000000
75%	30.000000	896.000000	5.500000	94.000000	15.000000
max	38.000000	7664.000000	12.500000	264.000000	75.000000

Summary statistics can only reveal so much



🏋 Exercise: A.1 - Effective EDA

The aim of this exercise is to perform Exploratory Data Analysis (EDA) of the Rossmann Kaggle dataset. You are provided with historical sales data for 1,115 Rossmann stores. The task is to forecast the "Sales" column for the test set. Note that some stores in the dataset were temporarily closed for refurbishment.

(Note: We will take a subset of the data with the features that we want to work upon. The data that we load is is not the complete data.)

Instructions

- Import the necessary libraries
- Load the data into a dataframe and observe the first few entries
- Summarize the statistics of the dataframe with the help of the helper function
- Perform basic feature engineering on the dataframe by performing the following
    - Create a column to represent the day of the week as a string, such as "Mon", "Tue" etc.