Data Visualization

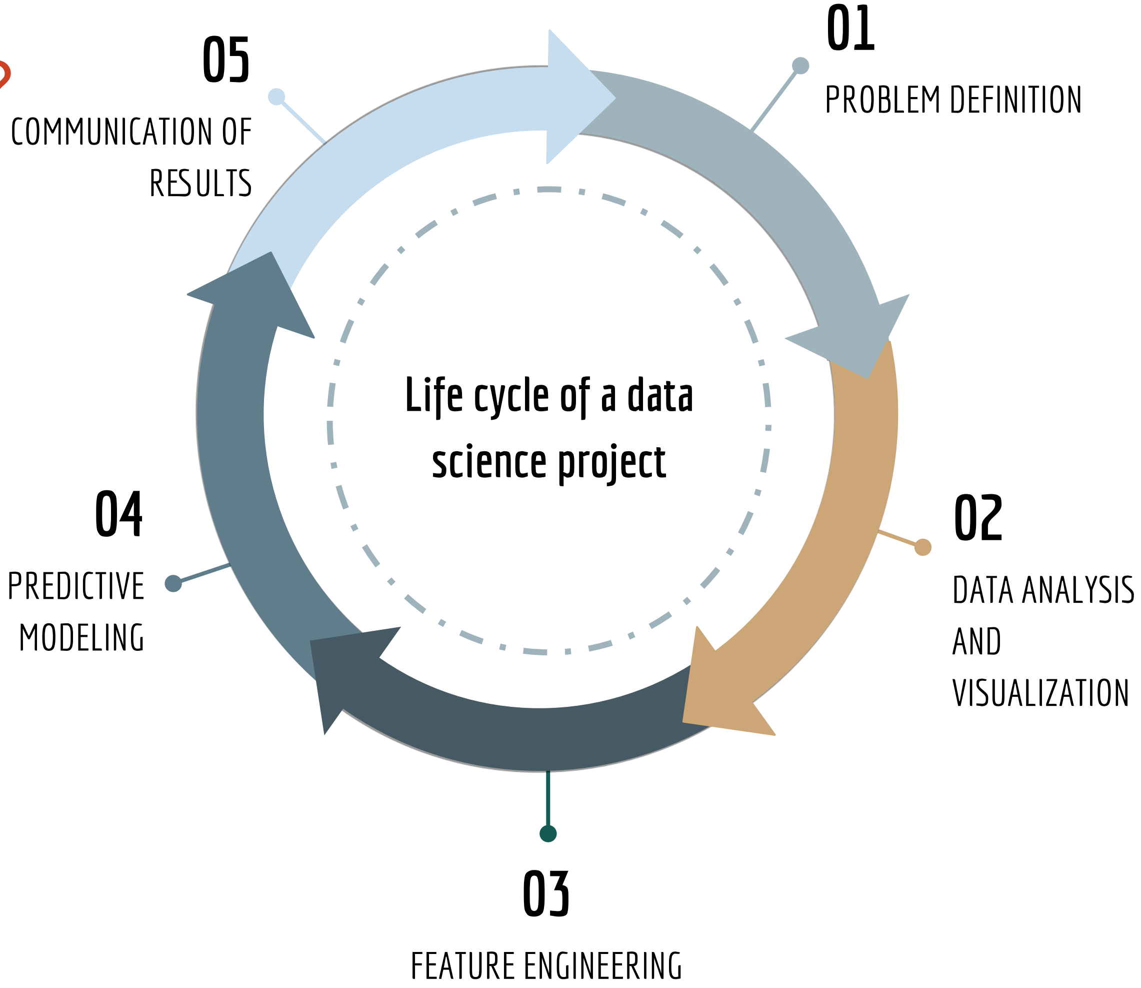
Basic Plots and Random Variables

CentraleDigitalLab@Nice

Data Analysis	Data Science	Machine Learning
Needs of **concrete questions**	Needs of a **problematic** in a domain	Needs of a **task** and a **dataset**.
Explains data to take a **future decision**	Aims to develop a **product based on data**	**Optimizes** a metric that measures **performance**
Guided by the data **analyst**	Guided by the **interpretation** of the data	Guided by the **model theory**
Detects **superficial patterns**	Highlights **deep patterns**	Detects **deep patterns**

What is Data Science?

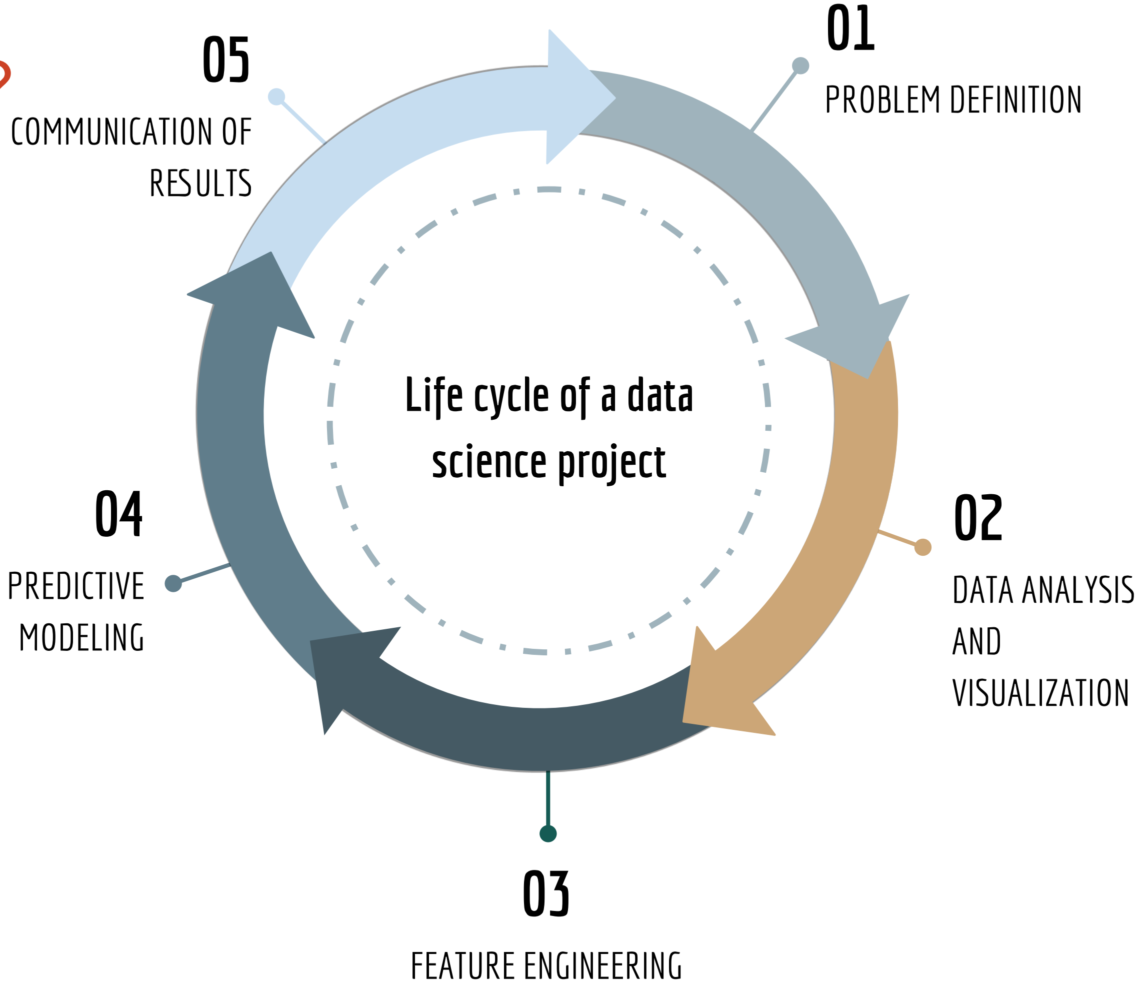
Data science is a discipline that aims to **develop a product based on data**.



What is Data Science?

Data science is a discipline that aims to **develop a product based on data**.

Uses approaches from the **data analysis** and **machine learning**.

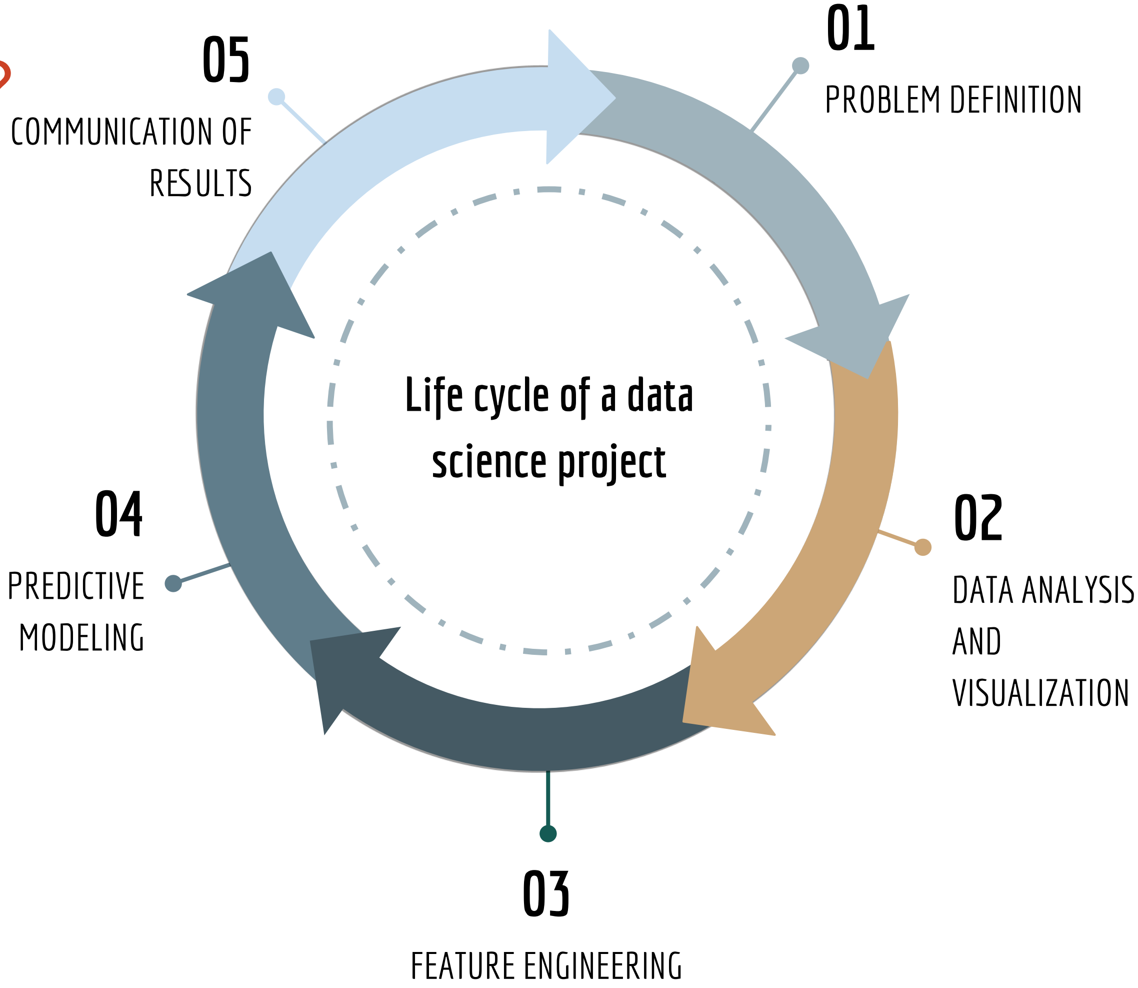


What is Data Science?

Data science is a discipline that aims to **develop a product based on data**.

Uses approaches from the **data analysis** and **machine learning**.

Visualization plays an important role on steps: **02**, **04** and **05.**



Learn ML without Data Visualization?

We **don’t know what to model** unless we are told to.

Learn ML without Data Visualization?

We **don’t know what to model** unless we are told to.

We **can’t understand** the impact of our **results**.

Learn ML without Data Visualization?

We **don’t know what to model** unless we are told to.

We **can’t understand** the impact of our **results**.

We **can’t detect bias**, **unfairness**, or **information filtering**.

Learn ML without Data Visualization?

We **don't know what to model** unless we are told to.

We **can't understand** the impact of our **results**.

We **can't detect bias**, **unfairness**, or **information filtering**.

It's necessary to understand the impact given by **business metrics**.

Learn ML without Data Visualization?

We **don’t know what to model** unless we are told to.

We **can’t understand** the impact of our **results**.

We **can’t detect bias**, **unfairness**, or **information filtering**.

It’s necessary to understand the impact given by **business metrics**.

We **waste** so much **time** and **effort** developing models that don’t answer our questions.

Learn Data Visualization without ML?

We are **limited to simple analysis**. Or we use models without understanding them.

Learn Data Visualization without ML?

We are **limited to simple analysis**. Or we use models without understanding them.

Use of machine learning **models** that are **inappropriate for our dataset**. For example models that don’t work well on categorical data.

Learn Data Visualization without ML?

We are **limited to simple analysis**. Or we use models without understanding them.

Use of machine learning **models** that are **inappropriate for our dataset**. For example models that don’t work well on categorical data.

We **spend so much time** optimizing a model since we don’t know how to properly do it.

Learn Data Visualization without ML?

We are **limited to simple analysis**. Or we use models without understanding them.

Use of machine learning **models** that are **inappropriate for our dataset**. For example models that don’t work well on categorical data.

We **spend so much time** optimizing a model since we don’t know how to properly do it.

We can only **detect superficial patterns**.

Data Visualization

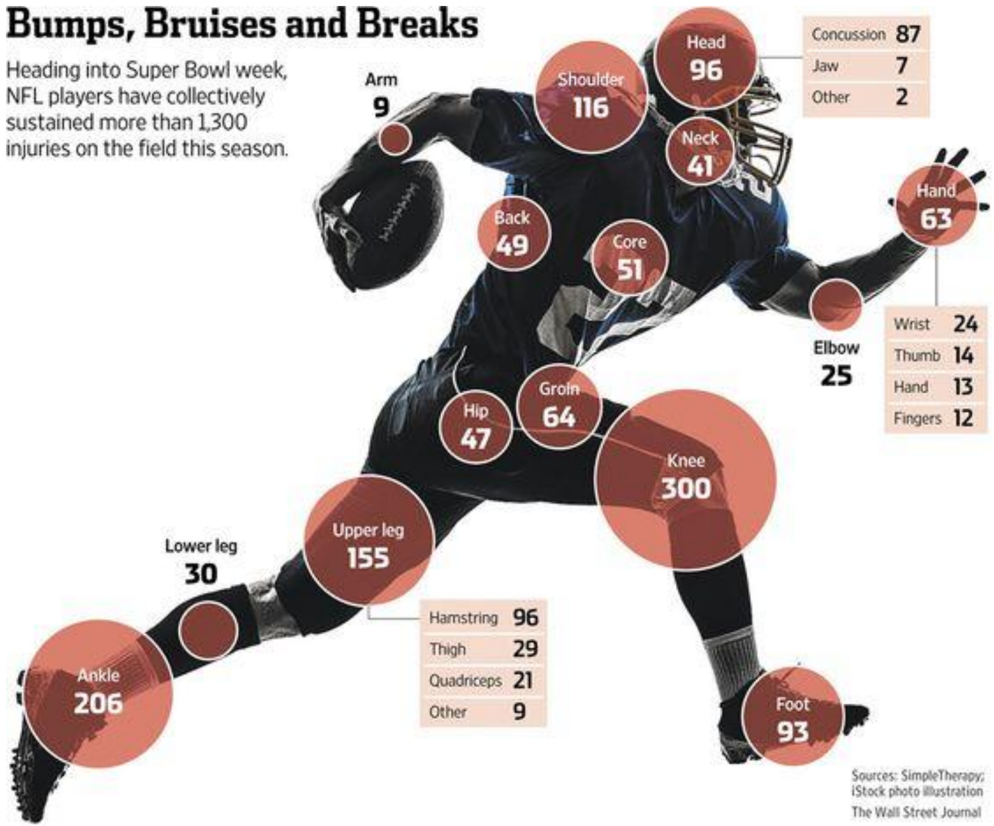
Data visualization is relevant in the data science process as it helps to:

Data Visualization

Data visualization is relevant in the data science process as it helps to:

- **Identify** relevant information and **properties in our dataset**.
- Detect patterns and **correlations between variables**.
- Experiment and **provide answers to hypothesis** during our research process.
- Recognize machine learning model **relevant features**.
- **Communicate results** to team members.

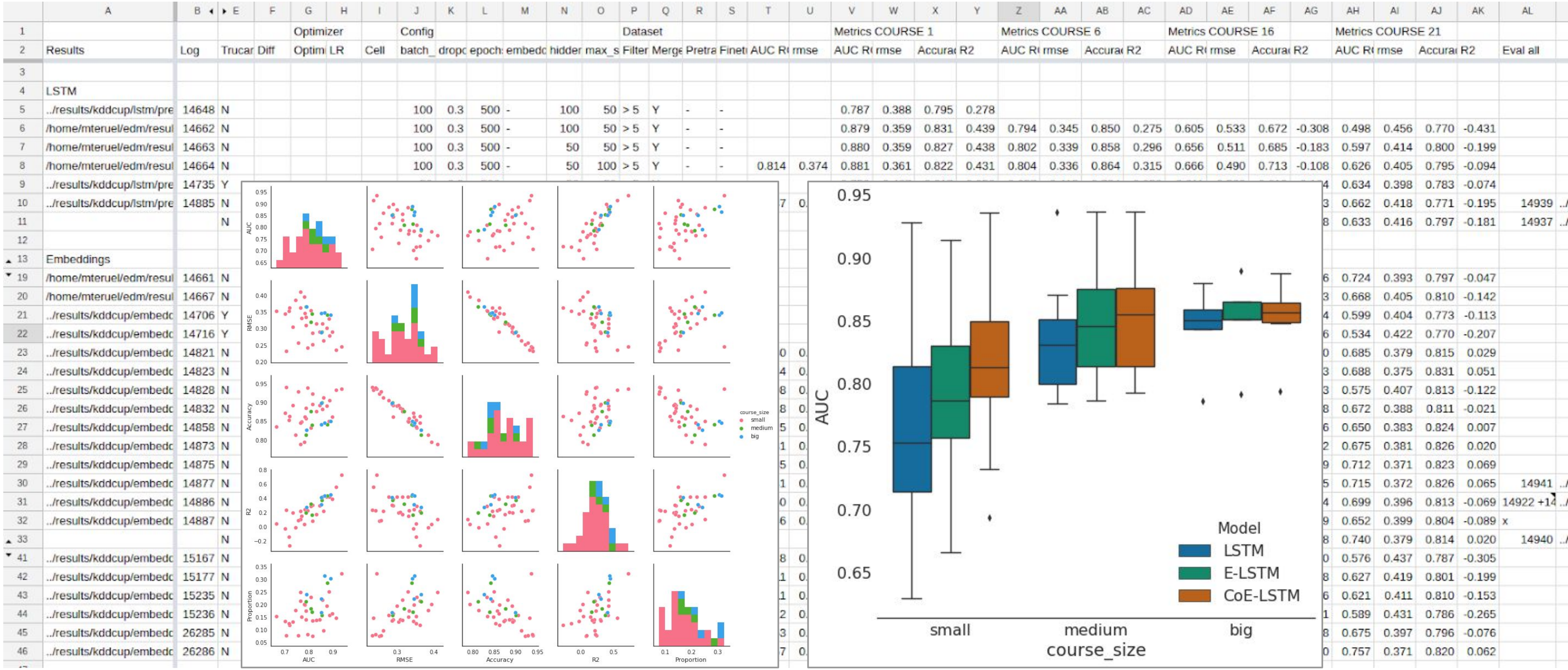
Some Examples



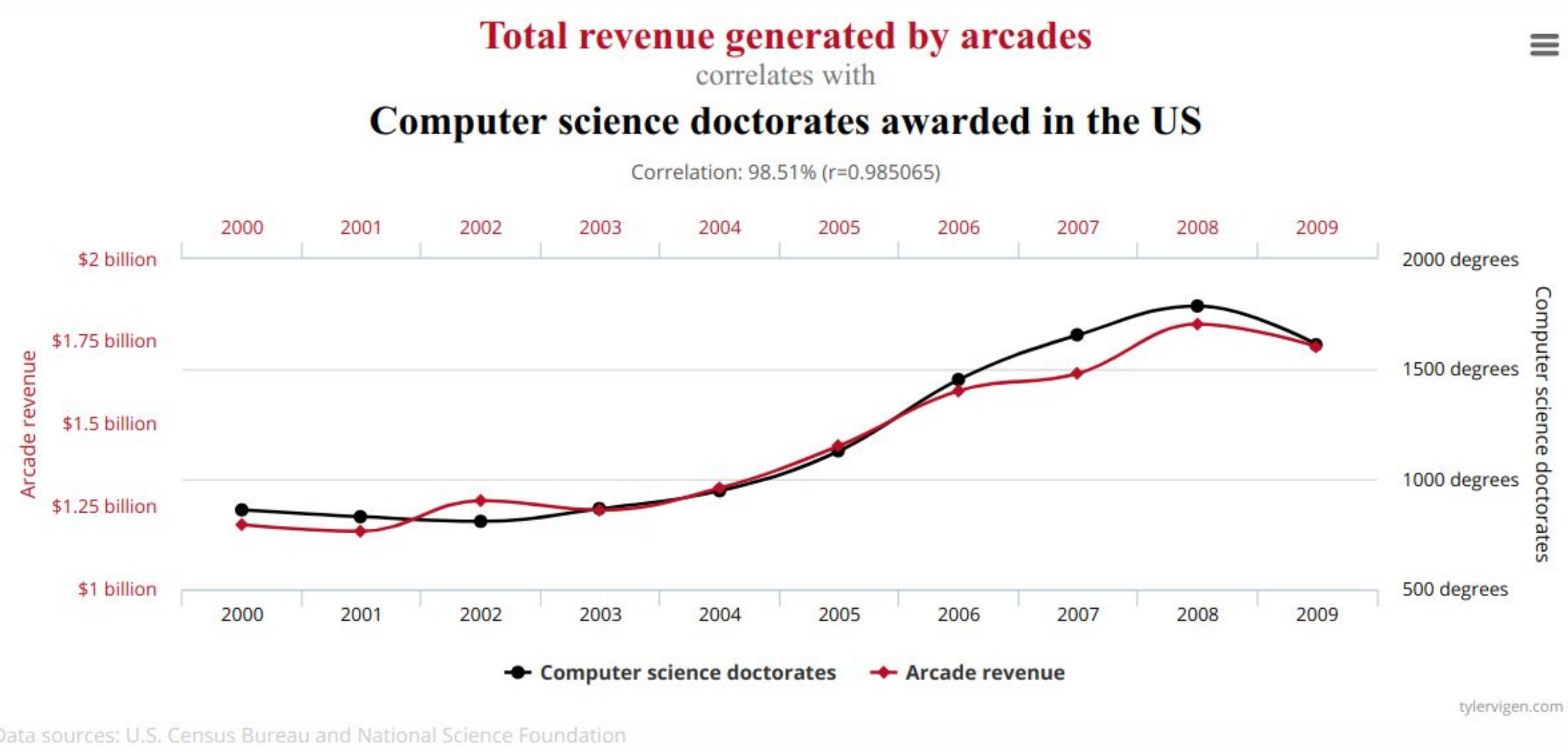
Some Examples

	A	B	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	
1					Optimizer			Config						Dataset						Metrics COURSE 1				Metrics COURSE 6				Metrics COURSE 16				Metrics COURSE 21					
2	Results	Log	Trucar	Diff	Optimi	LR	Cell	batch_	dropc	epoch:	embedc	hidder	max_s	Filter	Merge	Pretra	Fineti	AUC R(	rmse	AUC R(	rmse	Accura(	R2	AUC R(	rmse	Accura(	R2	AUC R(	rmse	Accura(	R2	AUC R(	rmse	Accura(	R2	Eval all	
3																																					
4	LSTM																																				
5	../results/kddcup/lstm/pre	14648	N					100	0.3	500	-	100	50	> 5	Y	-	-			0.787	0.388	0.795	0.278														
6	/home/mteruel/edm/resul	14662	N					100	0.3	500	-	100	50	> 5	Y	-	-			0.879	0.359	0.831	0.439	0.794	0.345	0.850	0.275	0.605	0.533	0.672	-0.308	0.498	0.456	0.770	-0.431		
7	/home/mteruel/edm/resul	14663	N					100	0.3	500	-	50	50	> 5	Y	-	-			0.880	0.359	0.827	0.438	0.802	0.339	0.858	0.296	0.656	0.511	0.685	-0.183	0.597	0.414	0.800	-0.199		
8	/home/mteruel/edm/resul	14664	N					100	0.3	500	-	50	100	> 5	Y	-	-	0.814	0.374	0.881	0.361	0.822	0.431	0.804	0.336	0.864	0.315	0.666	0.490	0.713	-0.108	0.626	0.405	0.795	-0.094		
9	../results/kddcup/lstm/pre	14735	Y					50	0.3	500	-	50	50	> 5	Y	-	-			0.759	0.467	0.617	0.050	0.657	0.415	0.784	-0.053	0.611	0.500	0.613	-0.164	0.634	0.398	0.783	-0.074		
10	../results/kddcup/lstm/pre	14885	N					50	0.3	500	-	50	50	N	Y	-	-	0.837	0.335	0.871	0.341	0.855	0.450	0.842	0.290	0.895	0.369	0.741	0.386	0.796	0.153	0.662	0.418	0.771	-0.195	14939	../
11			N					100	0.3	500	-	50	100	N	Y	-	-			0.871	0.339	0.851	0.458	0.837	0.289	0.896	0.375	0.748	0.390	0.800	0.138	0.633	0.416	0.797	-0.181	14937	../
12																																					
13	Embeddings																																				
19	/home/mteruel/edm/resul	14661	N					100	0.3	500	50	50	100	> 5	N	N				0.885	0.352	0.836	0.457	0.814	0.335	0.856	0.326	0.703	0.466	0.723	0.006	0.724	0.393	0.797	-0.047		
20	/home/mteruel/edm/resul	14667	N					100	0.3	500	50	50	50	> 5	N	N				0.880	0.359	0.829	0.439	0.813	0.332	0.867	0.330	0.687	0.478	0.705	-0.053	0.668	0.405	0.810	-0.142		
21	../results/kddcup/embedc	14706	Y					50	0.2	500	50	50	20	> 5	N	Y	Y			0.728	0.481	0.614	-0.010	0.683	0.390	0.798	0.069	0.641	0.467	0.687	0.004	0.599	0.404	0.773	-0.113		
22	../results/kddcup/embedc	14716	Y					100	0.3	500	50	50	50	> 5	Y	N				0.756	0.459	0.645	0.079	0.674	0.430	0.746	-0.153	0.649	0.462	0.673	0.036	0.534	0.422	0.770	-0.207		
23	../results/kddcup/embedc	14821	N					100	0.3	500	50	50	100	> 5	N	N		0.830	0.363	0.884	0.357	0.830	0.443	0.810	0.334	0.868	0.319	0.740	0.455	0.727	0.050	0.685	0.379	0.815	0.029		
24	../results/kddcup/embedc	14823	N					100	0.3	500	50	50	100	> 5	Y	N		0.834	0.361	0.884	0.358	0.825	0.442	0.813	0.333	0.865	0.322	0.731	0.456	0.715	0.043	0.688	0.375	0.831	0.051		
25	../results/kddcup/embedc	14828	N					50	0.2	500	50	100	20	> 5	Y	Y	Y	0.808	0.378	0.871	0.365	0.816	0.417	0.801	0.344	0.846	0.279	0.701	0.479	0.711	-0.053	0.575	0.407	0.813	-0.122		
26	../results/kddcup/embedc	14832	N					50	0.3	500	50	100	200	> 5	Y	Y	Y	0.818	0.370	0.879	0.359	0.825	0.436	0.788	0.340	0.861	0.296	0.707	0.471	0.711	-0.018	0.672	0.388	0.811	-0.021		
27	../results/kddcup/embedc	14858	N					50	0.3	500	50	100	200	> 5	Y	Y	N	0.825	0.365	0.875	0.365	0.827	0.420	0.806	0.338	0.854	0.302	0.714	0.449	0.739	0.076	0.650	0.383	0.824	0.007		
28	../results/kddcup/embedc	14873	N					50	0.3	500	20	100	200	> 5	Y	Y	Y	0.831	0.363	0.879	0.361	0.827	0.433	0.815	0.335	0.857	0.316	0.733	0.438	0.742	0.122	0.675	0.381	0.826	0.020		
29	../results/kddcup/embedc	14875	N					50	0.3	500	20	100	200	> 5	Y	Y	N	0.835	0.362	0.880	0.361	0.822	0.432	0.815	0.340	0.846	0.293	0.722	0.445	0.736	0.089	0.712	0.371	0.823	0.069		
30	../results/kddcup/embedc	14877	N					50	0.3	500	20	50	200	> 5	Y	Y	N	0.841	0.360	0.880	0.364	0.818	0.423	0.819	0.334	0.859	0.320	0.753	0.432	0.735	0.145	0.715	0.372	0.826	0.065	14941	../
31	../results/kddcup/embedc	14886	N					100	0.3	500	50	50	100	N	Y	N		0.850	0.330	0.887	0.338	0.853	0.461	0.850	0.291	0.895	0.366	0.783	0.379	0.807	0.184	0.699	0.396	0.813	-0.069	14922 +14	../
32	../results/kddcup/embedc	14887	N					50	0.3	500	20	100	200	N	Y	Y	Y	0.846	0.328	0.879	0.339	0.852	0.458	0.843	0.289	0.896	0.373	0.788	0.371	0.820	0.219	0.652	0.399	0.804	-0.089	x	
33			N					50	0.3	500	20	50	200	N	Y	Y	N			0.881	0.339	0.851	0.456	0.843	0.290	0.892	0.366	0.804	0.362	0.831	0.258	0.740	0.379	0.814	0.020	14940	../
41	../results/kddcup/embedc	15167	N		adam	0.01	gru	100	0.3	500	20	50	200	N	Y	Y	Y	0.818	0.343	0.890	0.334	0.854	0.472	0.839	0.294	0.896	0.349	0.732	0.391	0.807	0.130	0.576	0.437	0.787	-0.305		
42	../results/kddcup/embedc	15177	N		adam	??	gru	100	0.3	500	20	50	200	N	Y	Y	Y	0.811	0.345	0.886	0.336	0.858	0.466	0.823	0.304	0.890	0.307	0.683	0.438	0.773	-0.088	0.627	0.419	0.801	-0.199		
43	../results/kddcup/embedc	15235	N		adam	0.01	lstm	100	0.3	500	50	50	100	N	Y	N		0.811	0.347	0.882	0.345	0.841	0.439	0.814	0.304	0.892	0.305	0.684	0.425	0.799	-0.026	0.621	0.411	0.810	-0.153		
44	../results/kddcup/embedc	15236	N		adam	0.01	lstm	50	0.3	500	20	100	200	N	Y	Y	Y	0.812	0.345	0.884	0.340	0.850	0.452	0.826	0.300	0.892	0.325	0.700	0.405	0.800	0.071	0.589	0.431	0.786	-0.265		
45	../results/kddcup/embedc	26285	N					100	0.3	500	20	20	300	N	Y	N		0.853	0.325	0.881	0.335	0.854	0.469	0.841	0.291	0.895	0.362	0.790	0.369	0.814	0.228	0.675	0.397	0.796	-0.076		
46	../results/kddcup/embedc	26286	N					100	0.3	500	20	20	300	N	Y	Y	Y	0.857	0.322	0.883	0.334	0.857	0.474	0.845	0.288	0.895	0.375	0.783	0.364	0.826	0.250	0.757	0.371	0.820	0.062		

Some Examples



Some Examples



Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

Intuitively, a r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

Intuitively, a r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

The random variables can be of different types:

- Numerical
  - Continuous
  - Discrete (Infinite or finite set of numerable values)
- Categorical
- Ordinal

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

A r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

profile_gender	profile_age	profile_studies_level
Female	26	University
Male	29	University
Female	22	Secondary
Male	39	Postgraduate
Male	32	University
Male	25	Terciary
Male	33	University
Male	23	Terciary

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

A r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

Columns (Random Variables)

profile_gender	profile_age	profile_studies_level
Female	26	University
Male	29	University
Female	22	Secondary
Male	39	Postgraduate
Male	32	University
Male	25	Terciary
Male	33	University
Male	23	Terciary

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

A r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

Columns (Random Variables)

Rows (Elements of Ω)

profile_gender	profile_age	profile_studies_level
Female	26	University
Male	29	University
Female	22	Secondary
Male	39	Postgraduate
Male	32	University
Male	25	Terciary
Male	33	University
Male	23	Terciary

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

A r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

Columns (Random Variables)

Rows (Elements of Ω)

profile_gender	profile_age	profile_studies_level
Female	26	University
Male	29	University
Female	22	Secondary
Male	39	Postgraduate
Male	32	University
Male	25	Terciary
Male	33	University
Male	23	Terciary

Set of values of a r.v. (Range R)

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

A r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

X	Ω	R_X
Daily work hours	Software developers in France	1 - 24
Number of red blood cells	People with certain illness	Real numbers

Random Variable

A **random variable** (r.v.) X is a **function** X: Ω → R where **Ω is the state space** and **R** is the set of values that the variable can take called **Range**.

A r.v. is **equivalent to a column** of your dataset after applying 0 or more filters.

X	Ω	R_X
Daily work hours	Software developers in France	1 - 24
Number of red blood cells	People with certain illness	Real numbers

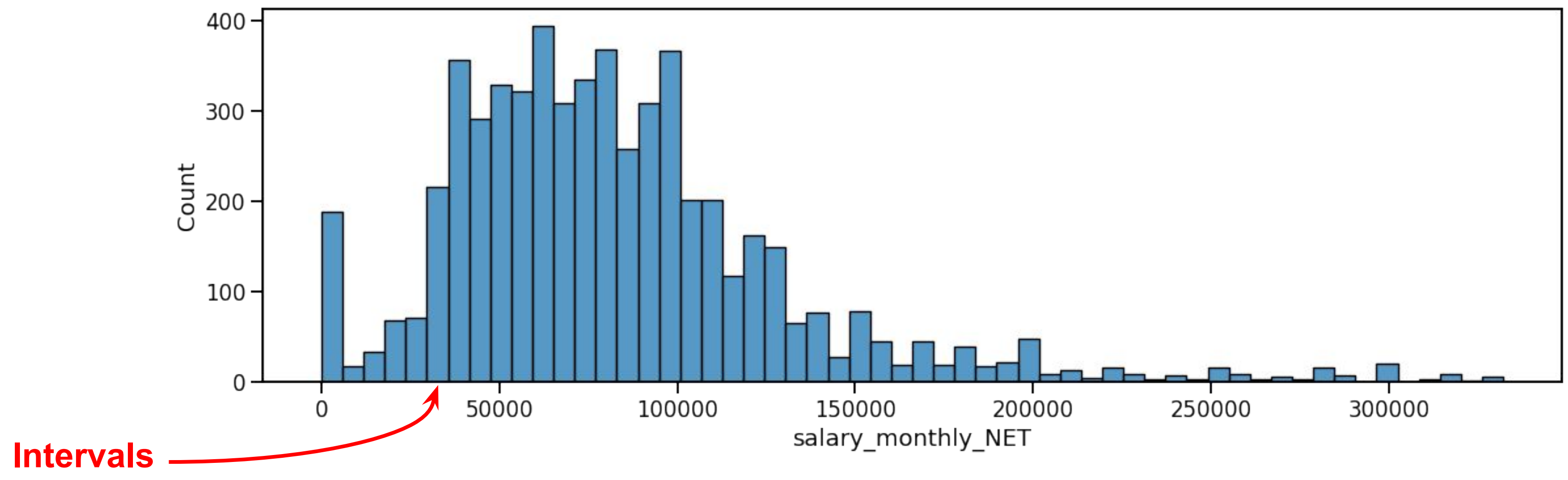
Can you give some other examples of RVs?

Basic Plots: Histogram

Mark **equally sized intervals** on a **horizontal** measurement axis. **Above each interval**, draw a rectangle whose **height is the corresponding count** (or relative frequency, density, percent, etc.).

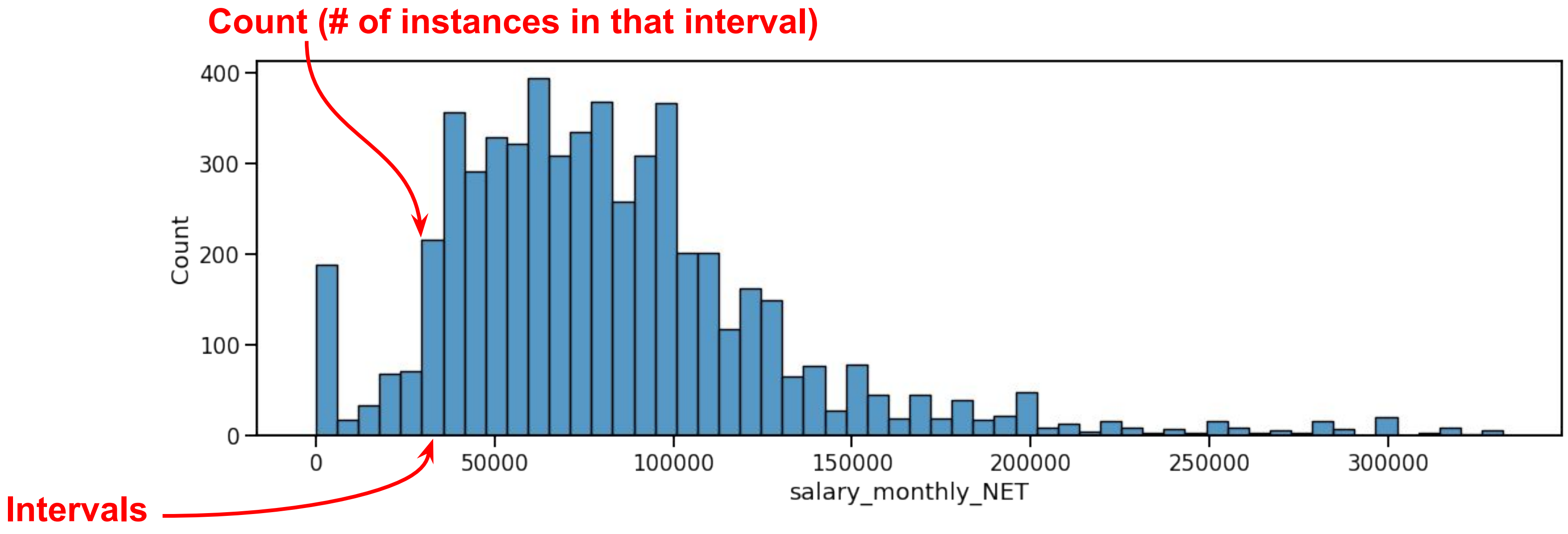
Basic Plots: Histogram

Mark **equally sized intervals** on a **horizontal** measurement axis. **Above each interval**, draw a rectangle whose **height is the corresponding count** (or relative frequency, density, percent, etc.).



Basic Plots: Histogram

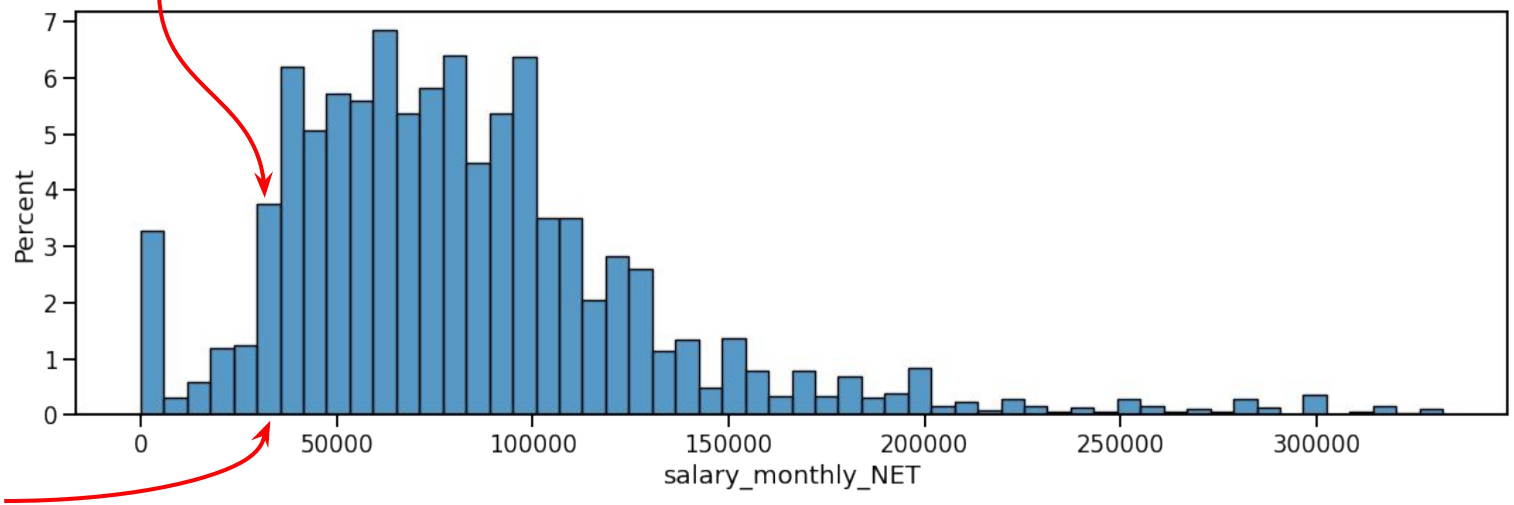
Mark **equally sized intervals** on a **horizontal** measurement axis. **Above each interval**, draw a rectangle whose **height is the corresponding count** (or relative frequency, density, percent, etc.).



Basic Plots: Histogram

Mark **equally sized intervals** on a **horizontal** measurement axis. **Above each interval**, draw a rectangle whose **height is the corresponding count** (or relative frequency, density, percent, etc.).

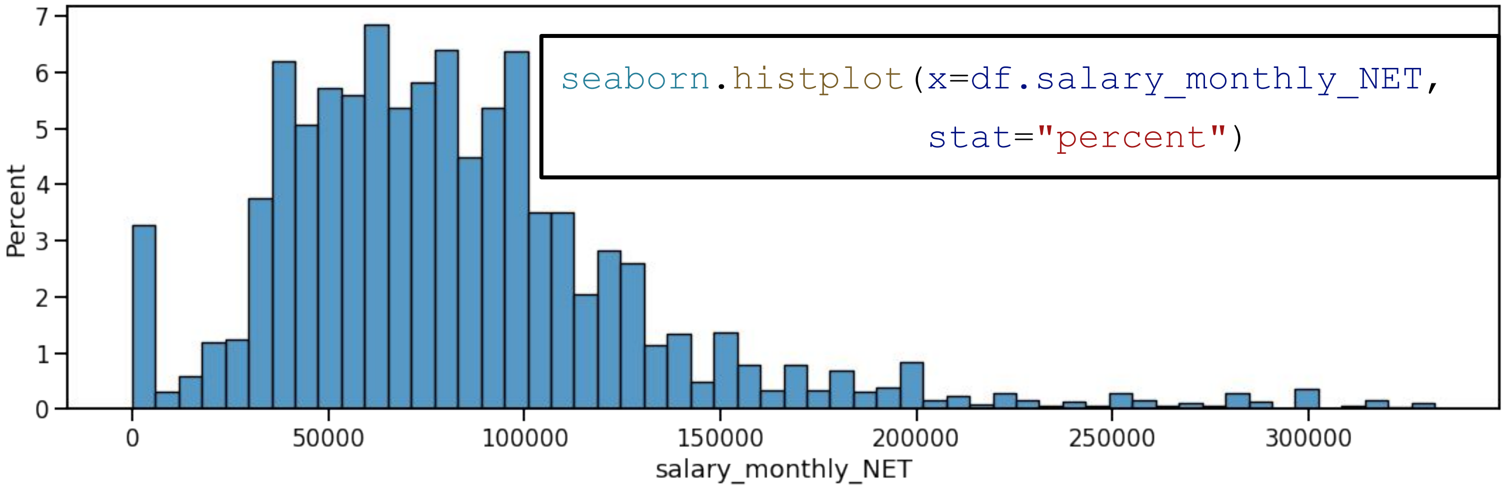
Percent (# of instances in that interval / total # of instances * 100)



Intervals

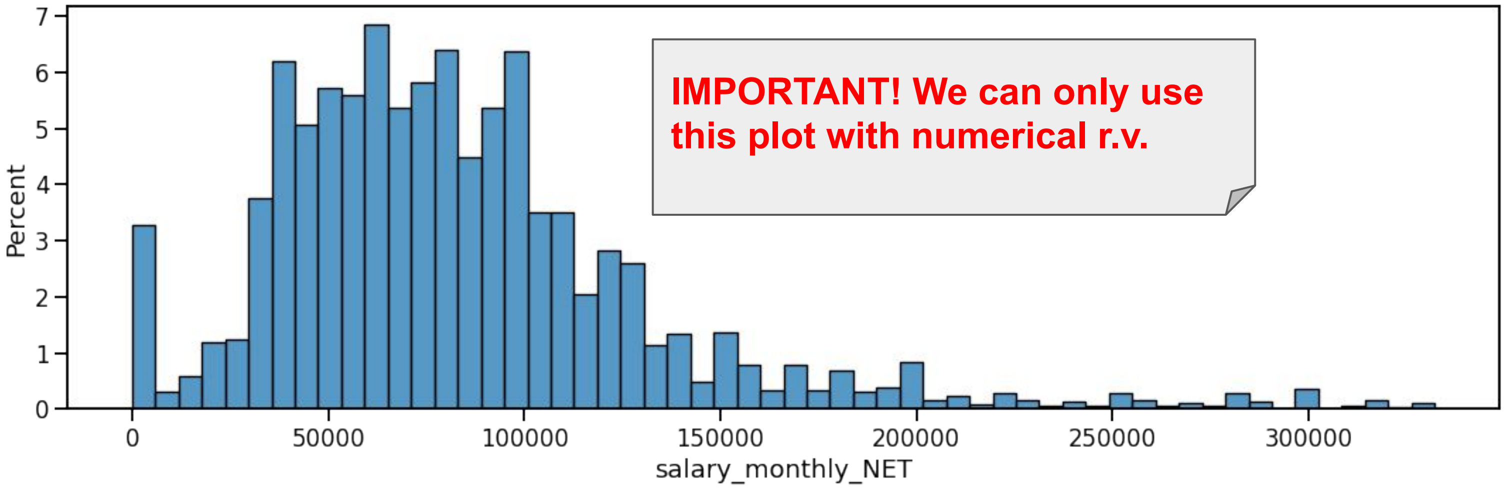
Basic Plots: Histogram

Mark **equally sized intervals** on a **horizontal** measurement axis. **Above each interval**, draw a rectangle whose **height is the corresponding count** (or relative frequency, density, percent, etc.).



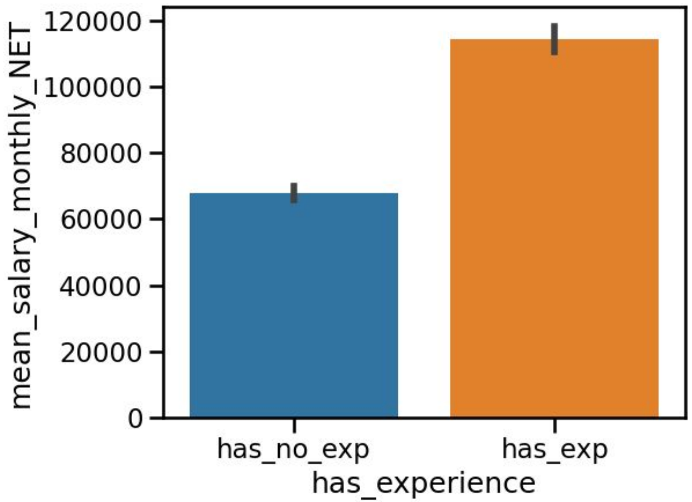
Basic Plots: Histogram

Mark **equally sized intervals** on a **horizontal** measurement axis. **Above each interval**, draw a rectangle whose **height is the corresponding count** (or relative frequency, density, percent, etc.).



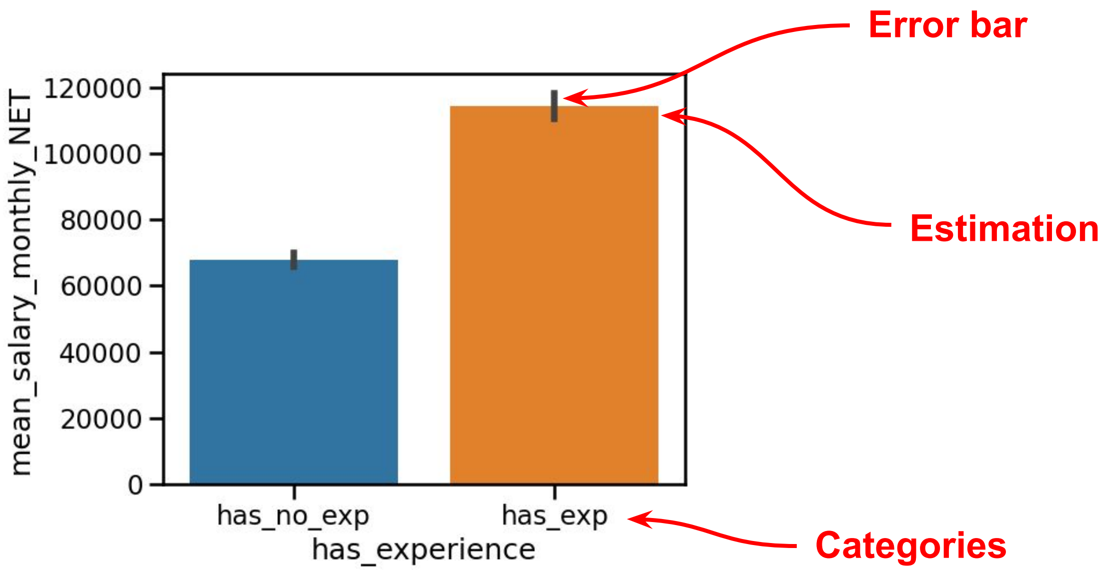
Basic Plots: Barplot

It represents an **estimate of central tendency for a numeric variable** with the height of each rectangle and provides some indication of the **uncertainty around that estimate** using error bars.



Basic Plots: Barplot

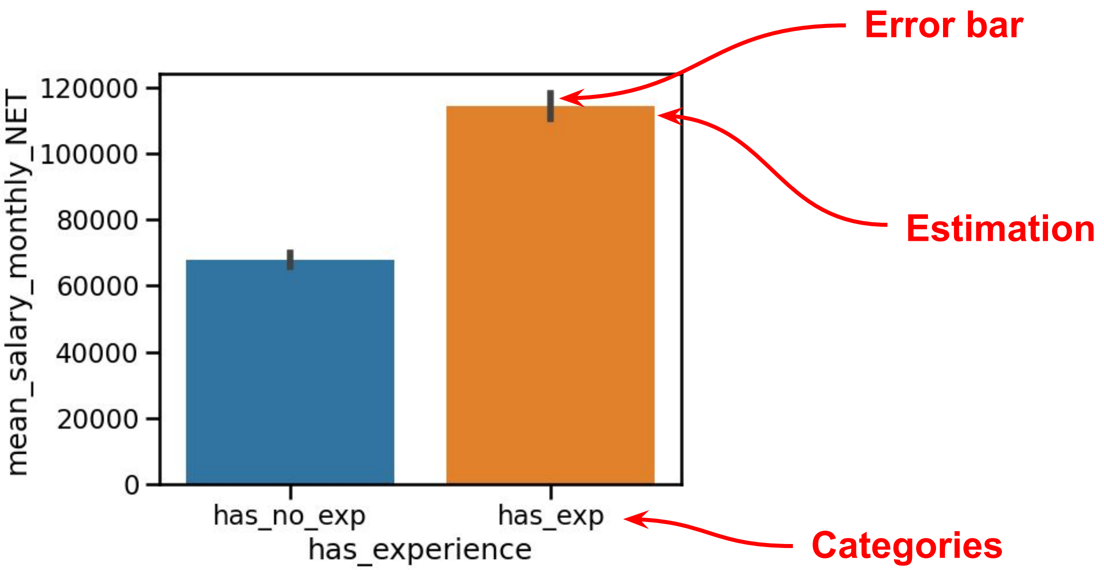
It represents an **estimate of central tendency for a numeric variable** with the height of each rectangle and provides some indication of the **uncertainty around that estimate** using error bars.



Basic Plots: Barplot

It represents an **estimate of central tendency for a numeric variable** with the height of each rectangle and provides some indication of the **uncertainty around that estimate** using error bars.

```
seaborn.barplot(
    data=df,
    x="has_experience",
    y="salary_monthly_NET")
```

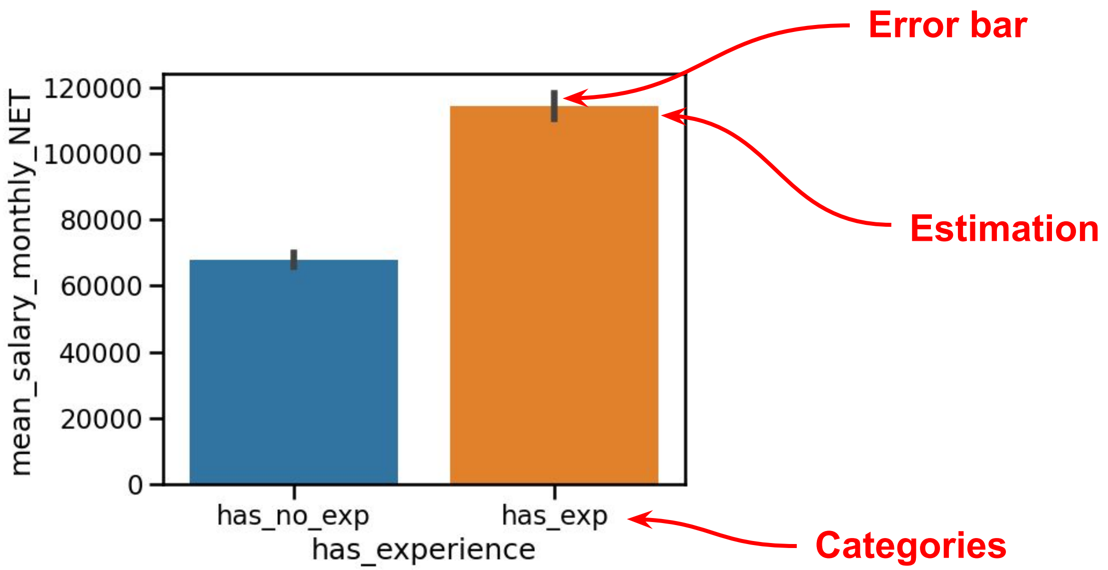


Basic Plots: Barplot

It represents an **estimate of central tendency for a numeric variable** with the height of each rectangle and provides some indication of the **uncertainty around that estimate** using error bars.

```
seaborn.barplot(
    data=df,
    x="has_experience",
    y="salary_monthly_NET")
```

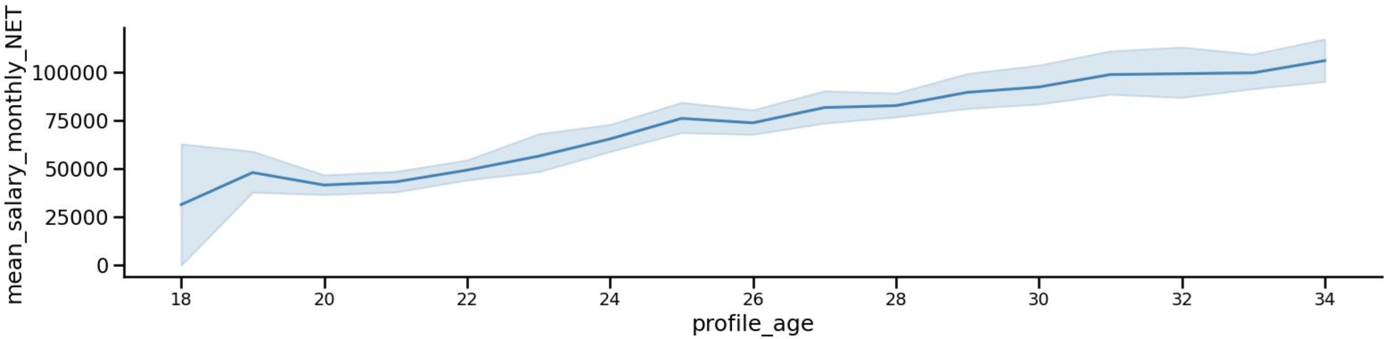
IMPORTANT! We can only use this plot with numerical r.v. in combination with a categorical one.



Basic Plots: Lineplot

It is useful when you want to **understand changes in one variable as a function of time**, or a similarly continuous variable.

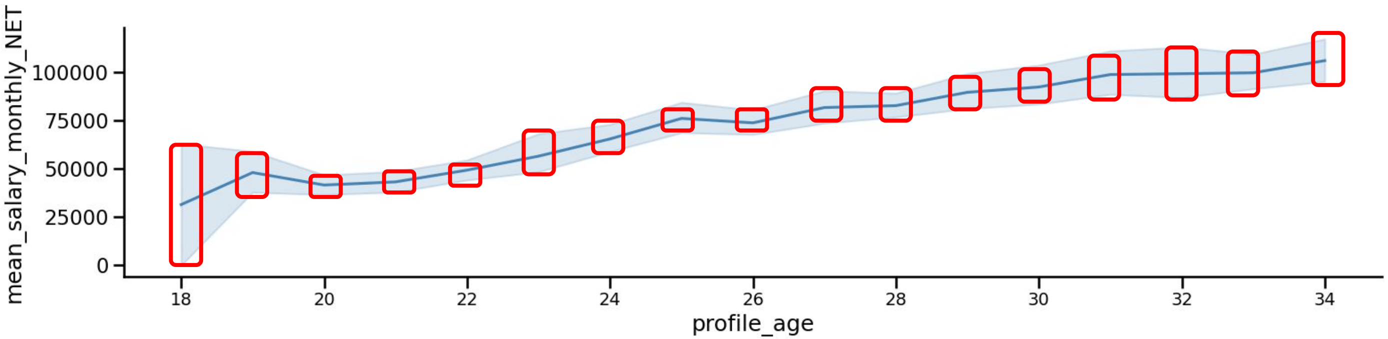
The plot **aggregates over multiple y values at each value of x** and shows an estimate of the central tendency and a confidence interval for that estimate.



Basic Plots: Lineplot

It is useful when you want to **understand changes in one variable as a function of time**, or a similarly continuous variable.

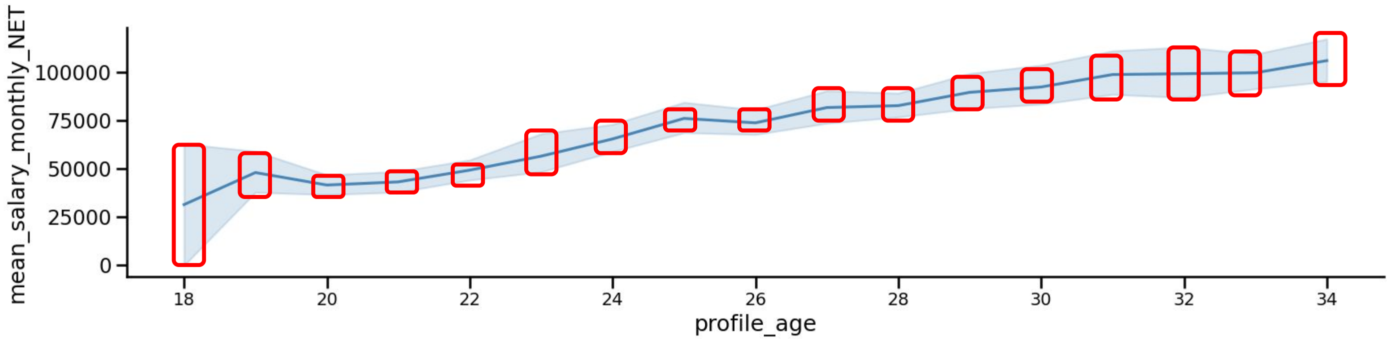
The plot **aggregates over multiple y values at each value of x** and shows an estimate of the central tendency and a confidence interval for that estimate.



Basic Plots: Lineplot

It is useful when you want to **understand changes in one variable as a function of time**, or a similarly continuous variable.

The plot **aggregates over multiple y values at each value of x** and shows an estimate of the central tendency and a confidence interval for that estimate.



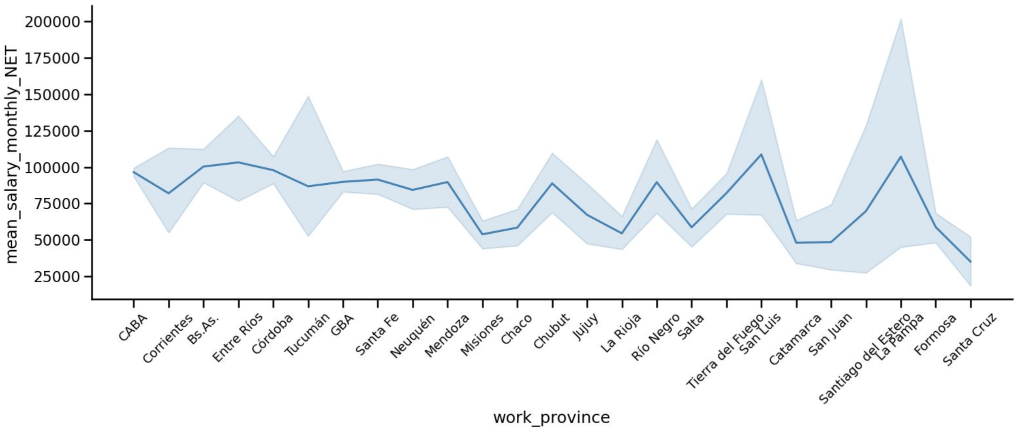

```
seaborn.lineplot(data=df, x="profile_age", y="salary_monthly_NET")
```

Basic Plots: Lineplot

It is useful when you want to **understand changes in one variable as a function of time**, or a similarly continuous variable.

The plot **aggregates over multiple y values at each value of x** and shows an estimate of the central tendency and a confidence interval for that estimate.

IMPORTANT! Don't use a categorical r.v. on the x axis.



Basic Plots: Boxplot

A boxplot is a standardized way of **displaying** a numerical r.v based on: the **minimum**, the **maximum**, the sample **median**, and the **first and third quartiles**.

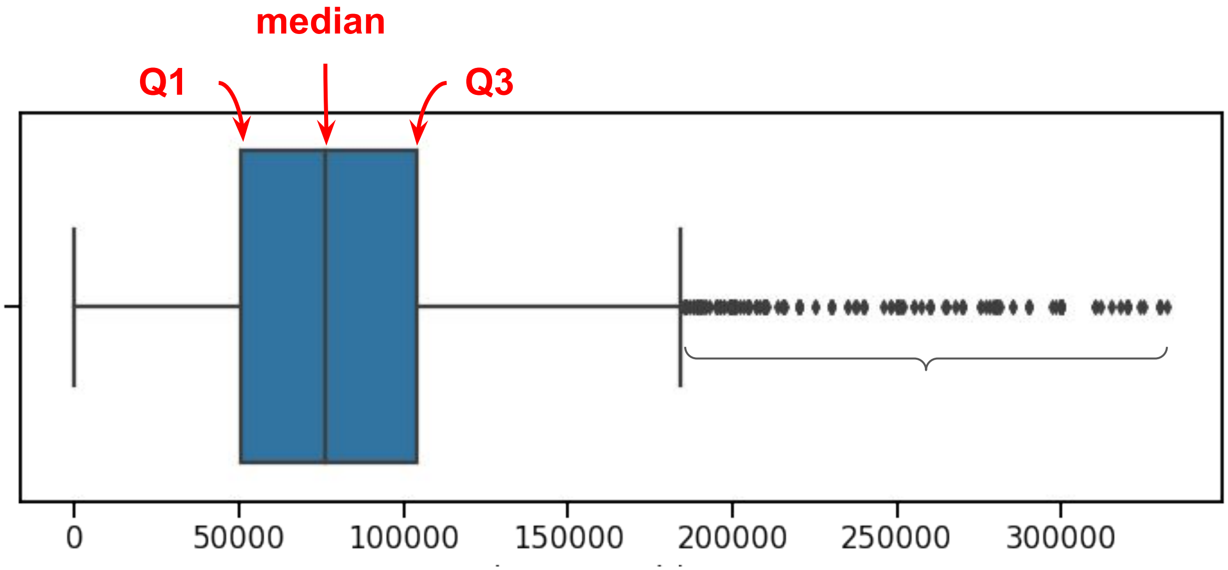
Basic Plots: Boxplot

A boxplot is a standardized way of **displaying** a numerical r.v based on: the **minimum**, the **maximum**, the sample **median**, and the **first and third quartiles**.

- Minimum: the lowest data point in the data set excluding any outliers
- Maximum: the highest data point in the data set excluding any outliers
- Median (50th percentile): the middle value in the data set
- First quartile (Q1 or 25th percentile): it is the median of the lower half of the dataset.
- Third quartile (Q3 or 75th percentile): it is the median of the upper half of the dataset.
- IRQ (Q3 - Q1): interquartile range

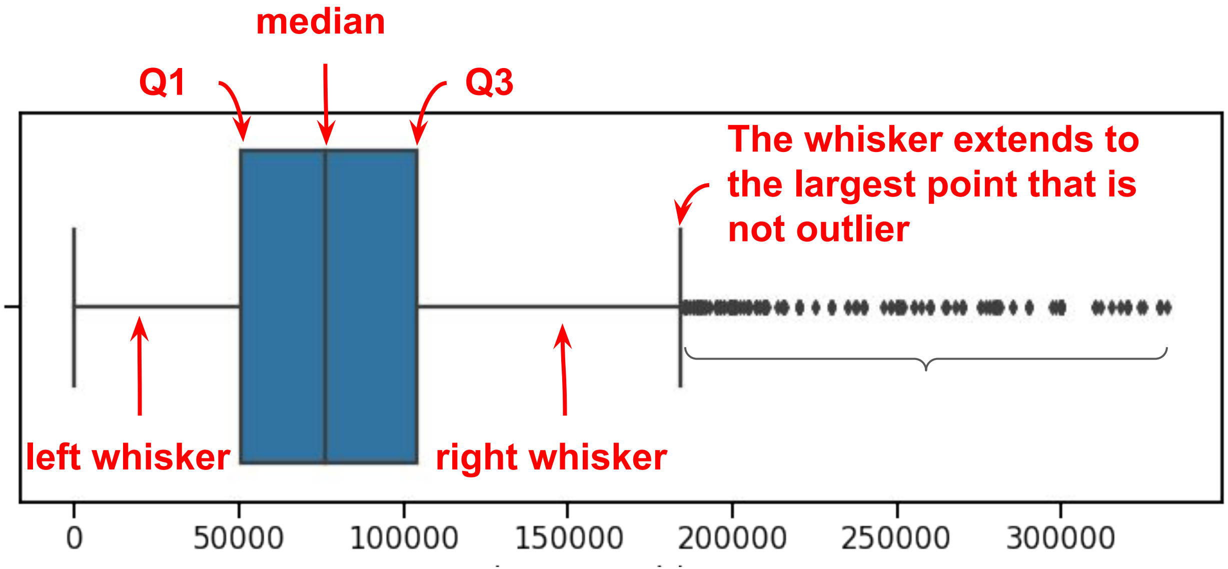
Basic Plots: Boxplot

The **box shows the quartiles** of the dataset while the **whiskers extend to show the rest of the distribution**, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.



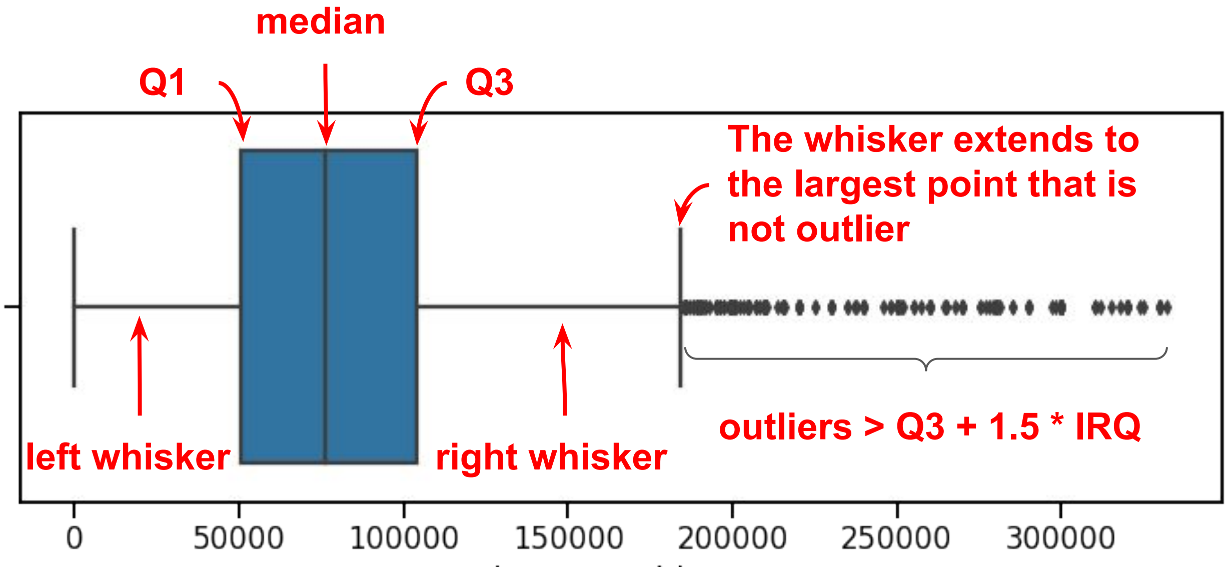
Basic Plots: Boxplot

The **box shows the quartiles** of the dataset while the **whiskers extend to show the rest of the distribution**, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.



Basic Plots: Boxplot

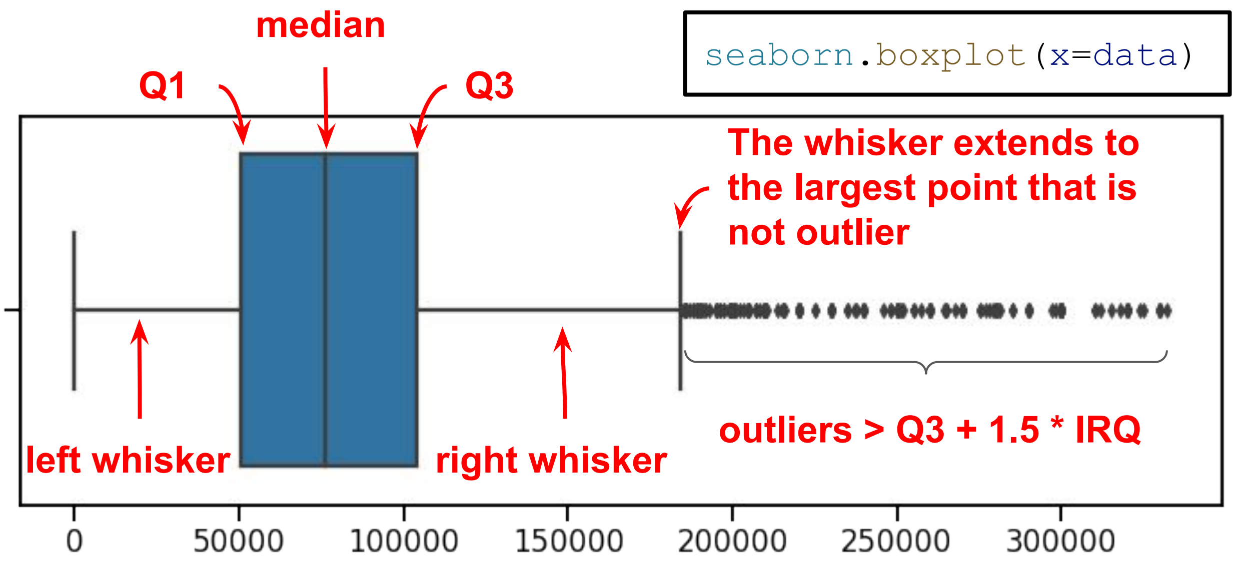
The **box shows the quartiles** of the dataset while the **whiskers extend to show the rest of the distribution**, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.



Basic Plots: Boxplot

The **box shows the quartiles** of the dataset while the **whiskers extend to show the rest of the distribution**, except for points that are determined to be "outliers" using a method that is a function of the inter-quartile range.

```
seaborn.boxplot(x=data)
```



Descriptive Statistics

- *Mean* $\bar{x} = \frac{1}{N}\sum_{i}^{N} x_i$
- *Variance* $v = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$
- *Percentil-k* $n = \left\lceil \frac{P}{100} \times N \right\rceil.$

- $median = x_{N/2}$
- $median = \frac{1}{2}(x_{\lfloor N/2 \rfloor} + x_{\lfloor N/2 \rfloor + 1})$

Probabilities

A probability **P** is a function **takes an state space Ω** and **returns a real number** between 0 and 1. At the same time, it has to **hold some properties**. Basically, for each subset A of Ω , P(A) is a number such as:

Probabilities

A probability **P** is a function **takes an state space Ω** and **returns a real number** between 0 and 1. At the same time, it has to **hold some properties**. Basically, for each subset A of Ω , P(A) is a number such as:

- $\mathbf{0} \leq \mathbf{P}(A) \leq 1$
- $\mathbf{P}(\boldsymbol{\Omega}) = 1$
- $\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B)$, for A and B disjoints
- $\mathbf{P}(\cup_i A_i) = \sum_i \mathbf{P}(A_i)$ for A_1, A_2 ,... disjoints

Probabilities

A probability **P** is a function **takes an state space Ω** and **returns a real number** between 0 and 1. At the same time, it has to **hold some properties**. Basically, for each subset A of Ω , P(A) is a number such as:

- $\mathbf{0} \leq \mathbf{P}(A) \leq 1$
- $\mathbf{P}(\boldsymbol{\Omega}) = 1$
- $\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B)$, for A and B disjoints
- $\mathbf{P}(\cup_i A_i) = \sum_i \mathbf{P}(A_i)$ for $A_1, A_2, \ldots$ disjoints

Events can be thought as **restrictions applied to one or several r.v.** Conditional probability between the two events is defined as:

$\mathbf{P}(A|B) = \mathbf{P}(A \text{ and } B) / P(B)$

$\mathbf{P}(A|B) = |A \text{ and } B| / |B|$

Probabilities

A probability **P** is a function **takes an state space Ω** and **returns a real number** between 0 and 1. At the same time, it has to **hold some properties**. Basically, for each subset A of Ω , P(A) is a number such as:

- $\mathbf{0} \leq \mathbf{P}(A) \leq 1$
- $\mathbf{P}(\boldsymbol{\Omega}) = 1$
- $\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B)$, for A and B disjoints
- $\mathbf{P}(\cup_i A_i) = \sum_i \mathbf{P}(A_i)$ for $A_1, A_2, \ldots$ disjoints

Events can be thought as **restrictions applied to one or several r.v.** Conditional probability between the two events is defined as:

$\mathbf{P}(A|B) = \mathbf{P}(A \text{ and } B) / P(B)$

$\mathbf{P}(A|B) = |A \text{ and } B| / |B|$

- A = { ω_i : salary_monthly_NET > avg(salary_monthly_NET) }
- B = { ω_i : profile_years_experience > 5 }

Common Operations on Dataframes

We can apply certain operations on a dataframe. The simplest ones are **projections** and **filterings**.

Common Operations on Dataframes

We can apply certain operations on a dataframe. The simplest ones are **projections** and **filterings**.

Projections: Put in brackets the name of the column we want to project.

```
df["profile_gender"], df["profile_age"], df[["profile_gender", "profile_age"]]
```

Common Operations on Dataframes

We can apply certain operations on a dataframe. The simplest ones are **projections** and **filterings**.

Projections: Put in brackets the name of the column we want to project.

```
df["profile_gender"], df["profile_age"], df[["profile_gender", "profile_age"]]
```

Filterings: Create a Pandas Series of booleans and give it as input to a dataframe of the same shape.

```
df[(df["profile_gender"] == "Male") &
    (df["profile_age"] < 30)]
```

Condition to filter

Demo with notebook

01_probability_and_basic_plots.ipynb

Removing Outliers

An outlier is **an observation point that is distant** from other observations. How to identify them?

Removing Outliers

An outlier is **an observation point that is distant** from other observations. How to identify them?

Intuition: With **data where you already know the distribution** (like people's ages), you can use common sense to find outliers that were incorrectly recorded. For example, you know that 356 is not a valid age, while 45 is.

Removing Outliers

An outlier is **an observation point that is distant** from other observations. How to identify them?

Intuition: With **data where you already know the distribution** (like people's ages), you can use common sense to find outliers that were incorrectly recorded. For example, you know that 356 is not a valid age, while 45 is.

Visualization: Looking at **variables together** can help you spot common-sense outliers. Say a study is using both people's ages and marital status to draw conclusions. **If you look at variables separately, you might miss outliers**. For example, "12 years old" isn't an outlier and "widow" isn't an outlier, but we know that a 12-year-old widow is likely an outlier.

Removing Outliers

An outlier is **an observation point that is distant** from other observations. How to identify them?

Percentiles: Using the **highest percentiles** we can check if they are **far apart from the rest of observations**. For example by calculating the percentiles .90, .95, or .99. You base on your knowledge in the domain.

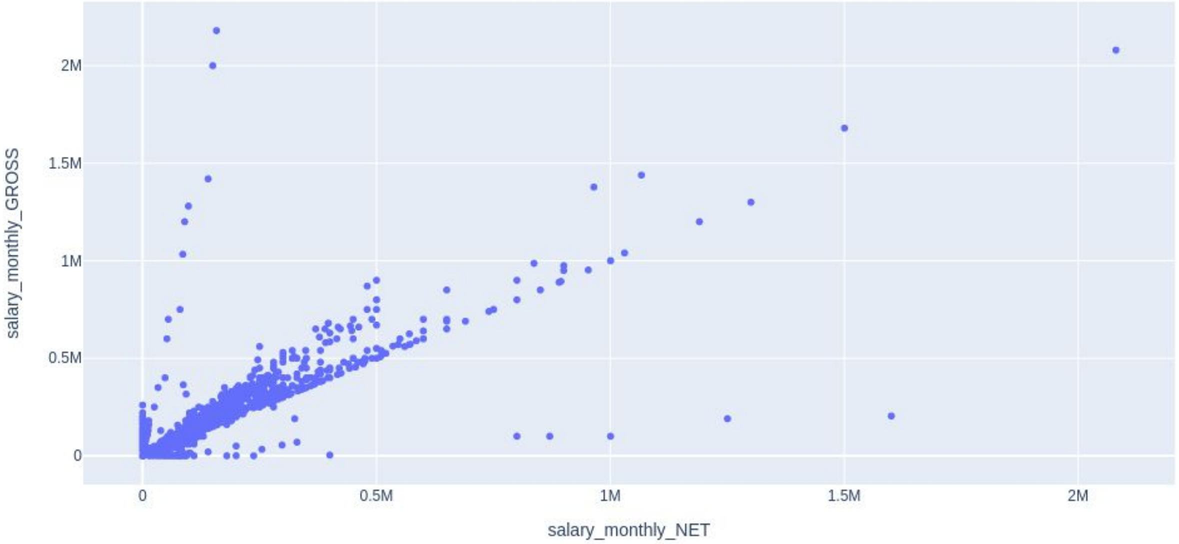
Boxplot: We can use the same approach used in boxplots by removing all those observations which are not in the interval **(Q1 - 1.5 * IRQ, Q3 + 1.5 * IRQ)**

Demo with notebook

02_descriptive_statistics.ipynb

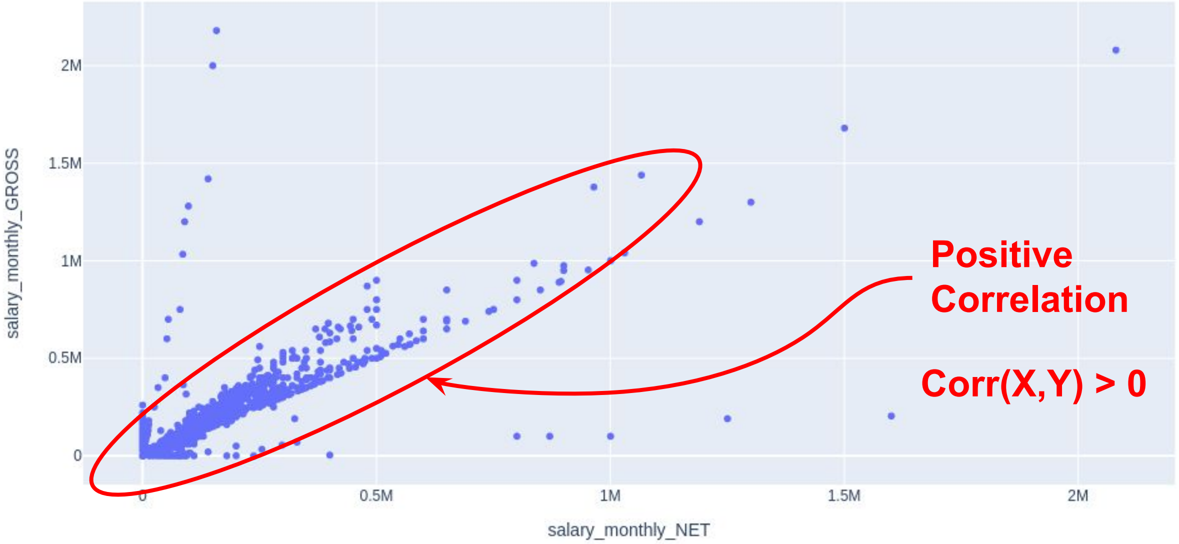
Two variables: Scatterplots

The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.



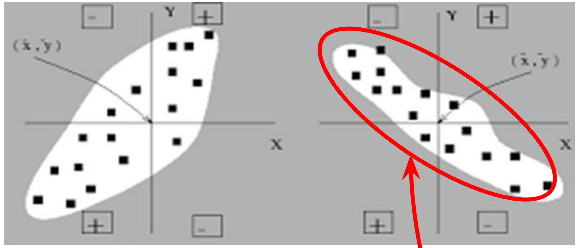
Two variables: Scatterplots

The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.



Positive Correlation

Corr(X,Y) > 0

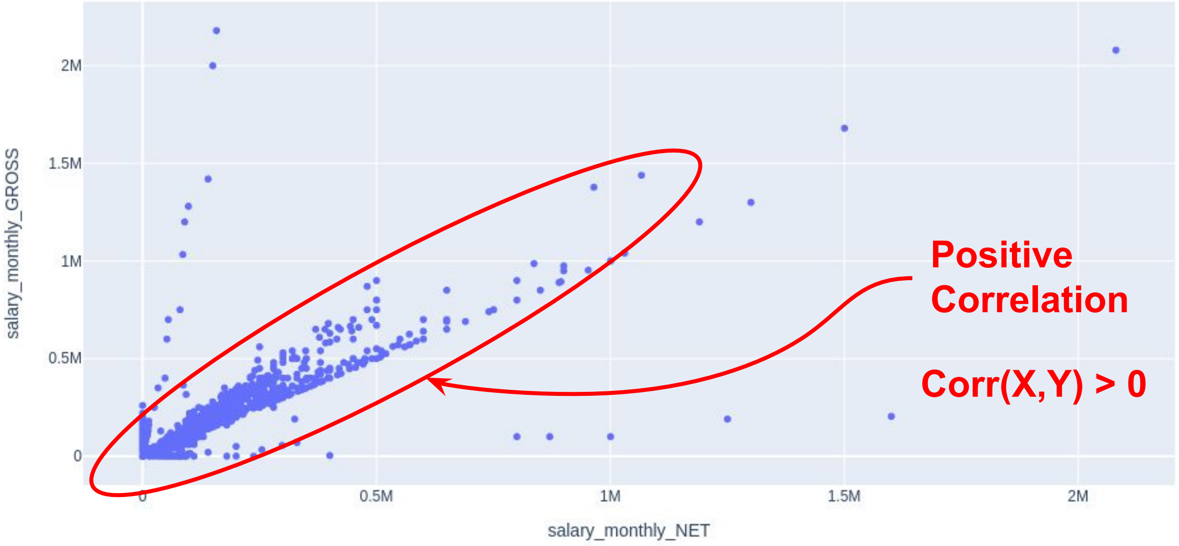


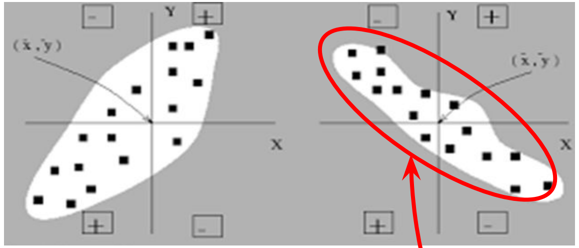
Negative Correlation

Corr(X,Y) < 0

Two variables: Scatterplots

The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.





Negative Correlation

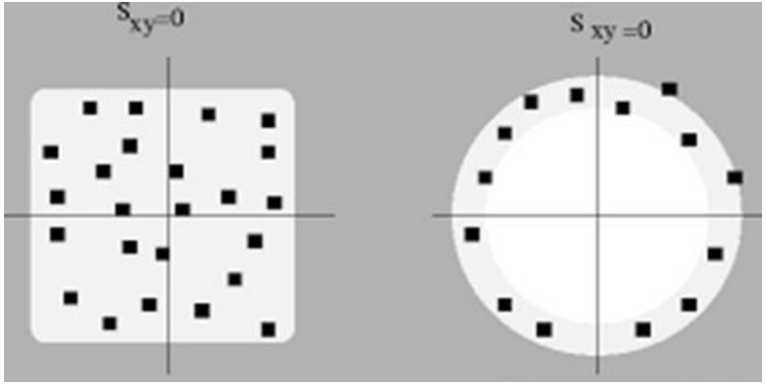
Corr(X,Y) < 0

IMPORTANT! We can only use this plot with numerical two continuous r.v.

- 1 <= Corr(X,Y) <= 1

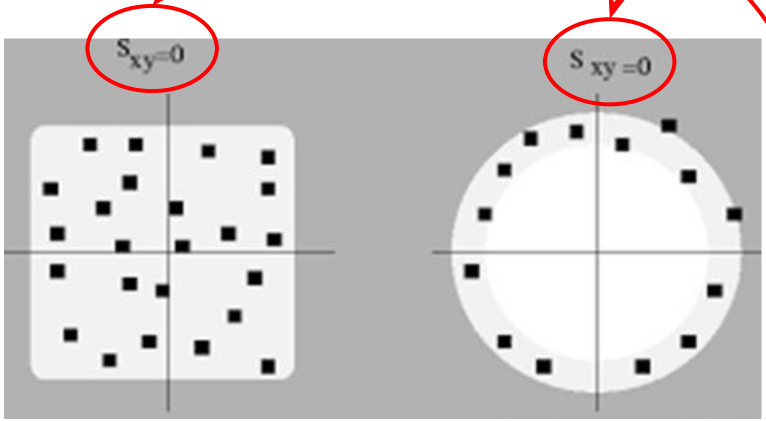
Two variables: Scatterplots

The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.



Two variables: Scatterplots

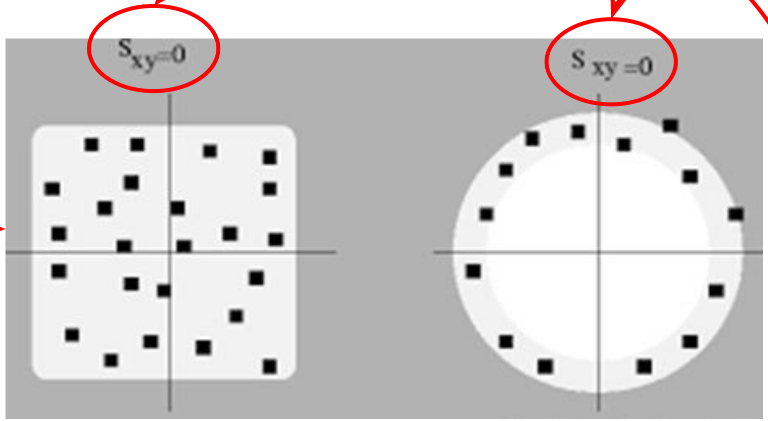
The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.



Both plots have linear correlation equal to 0

Two variables: Scatterplots

The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.

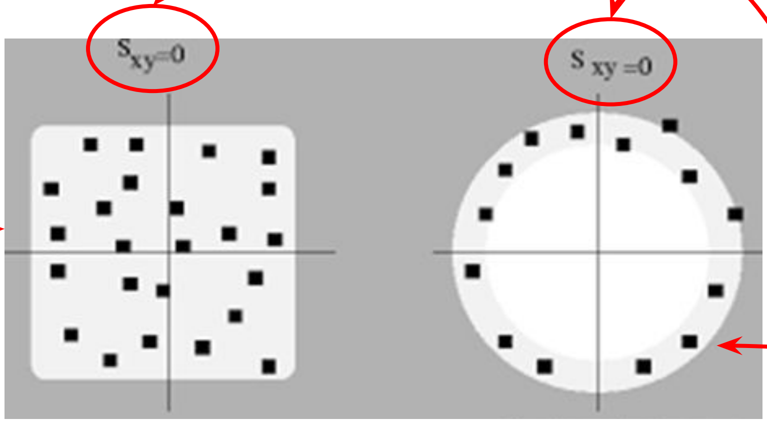


No correlation

Both plots have linear correlation equal to 0

Two variables: Scatterplots

The scatterplot shows the **relationship between two numerical r.v.** X and Y by mapping realizations of both variables into an 2D plot.



No correlation

Both plots have linear correlation equal to 0

Correlation but non-linear

Two variables: Scatterplots

- If $r = 1$, there's a **perfect positive correlation**. The coef. indicates a total dependence between the two variables called a direct relationship: when one of them increases, the other also does so in a constant proportion.

Two variables: Scatterplots

- If $r = 1$, there's a **perfect positive correlation**. The coef. indicates a total dependence between the two variables called a direct relationship: when one of them increases, the other also does so in a constant proportion.
- If $0 < r < 1$, there is a **positive correlation**.

Two variables: Scatterplots

- If $r = 1$, there's a **perfect positive correlation**. The coef. indicates a total dependence between the two variables called a direct relationship: when one of them increases, the other also does so in a constant proportion.
- If $0 < r < 1$, there is a **positive correlation**.
- If $r = 0$, there is **no linear relationship**. But this does not necessarily imply that the variables are independent: there **may still be nonlinear relationships** between the two variables.

Two variables: Scatterplots

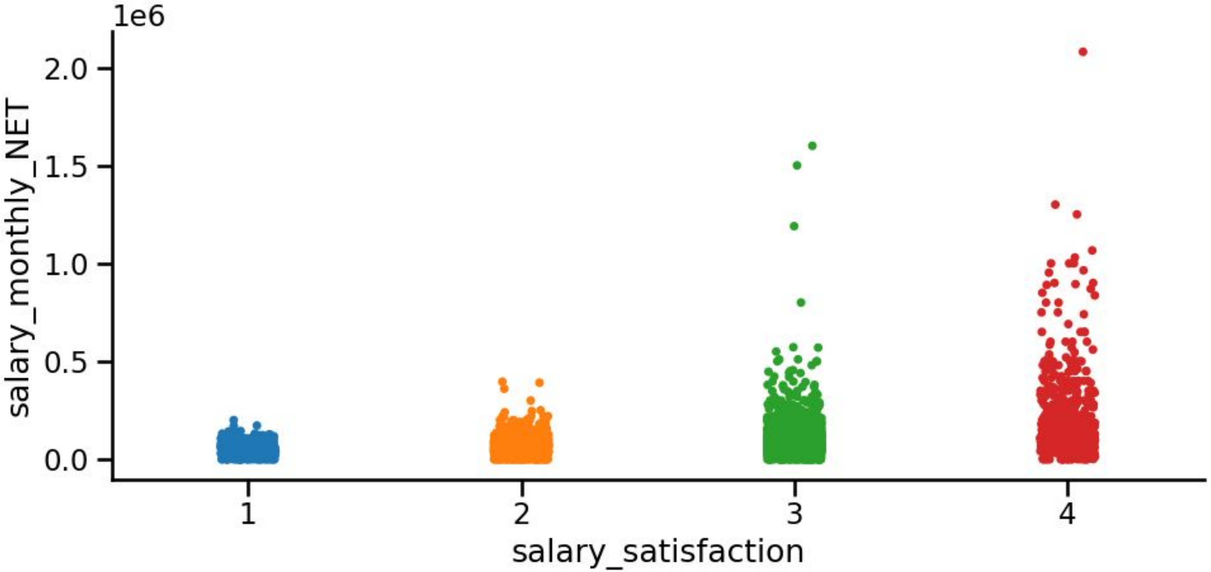
- If $r = 1$, there's a **perfect positive correlation**. The coef. indicates a total dependence between the two variables called a direct relationship: when one of them increases, the other also does so in a constant proportion.
- If $0 < r < 1$, there is a **positive correlation**.
- If $r = 0$, there is **no linear relationship**. But this does not necessarily imply that the variables are independent: there **may still be nonlinear relationships** between the two variables.
- If $-1 < r < 0$, there is a **negative correlation**.

Two variables: Scatterplots

- If *r* = 1, there's a **perfect positive correlation**. The coef. indicates a total dependence between the two variables called a direct relationship: when one of them increases, the other also does so in a constant proportion.
- If 0 < r < 1, there is a **positive correlation**.
- If r = 0, there is **no linear relationship**. But this does not necessarily imply that the variables are independent: there **may still be nonlinear relationships** between the two variables.
- If -1 < r < 0, there is a **negative correlation**.
- If r = -1, there is a **perfect negative correlation**. The coef. indicates a total dependence between the two variables called an inverse relationship: when one of them increases, the other decreases in constant proportion.

Two variables: Catplots

The catplots shows the **relationship between one categorical r.v.** X and **one numerical** Y.



Two variables: Catplots

The catplots shows the **relationship between one categorical r.v.** X and **one numerical** Y.

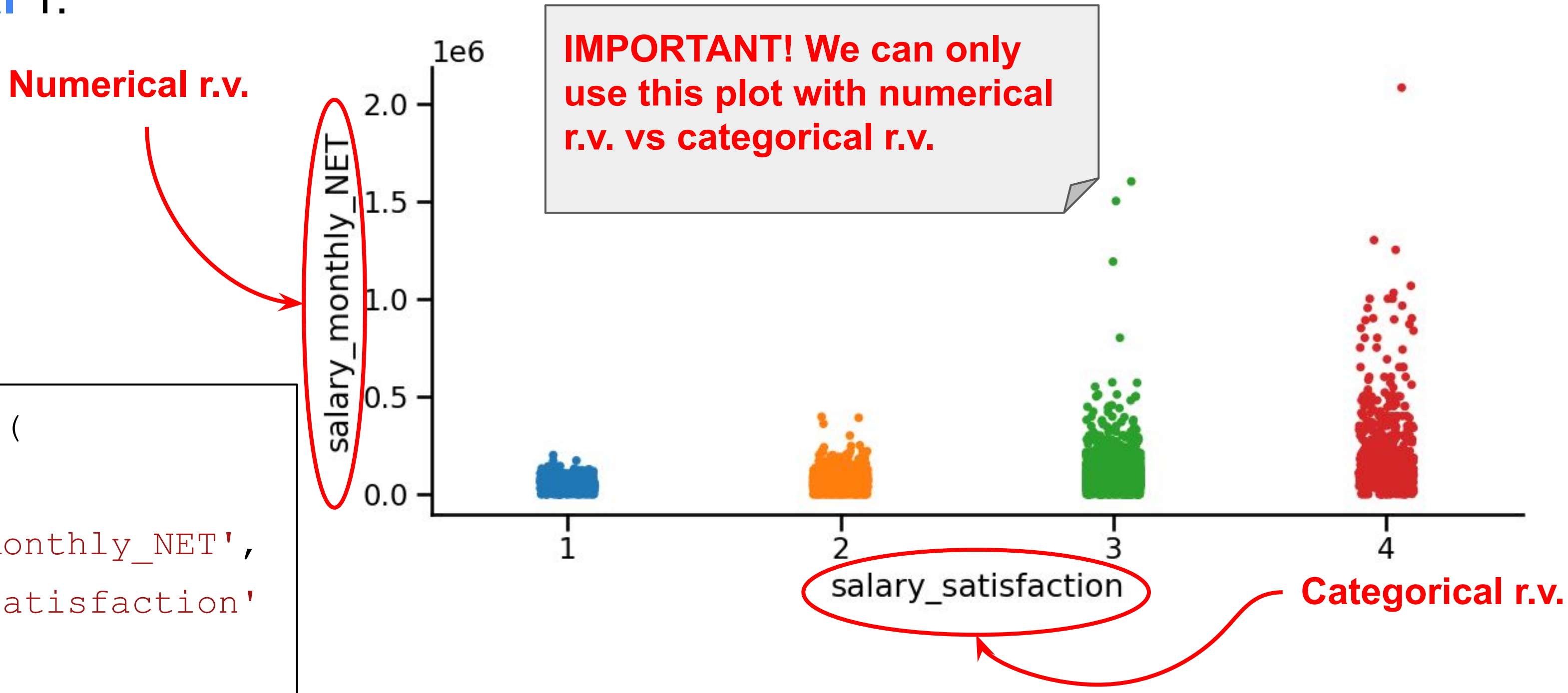
Two variables: Catplots

The catplots shows the **relationship between one categorical r.v.** X and **one numerical** Y.


```
sns.catplot(
data=df,
y='salary_monthly_NET',
x='salary_satisfaction'
)
```

Two variables: Catplots

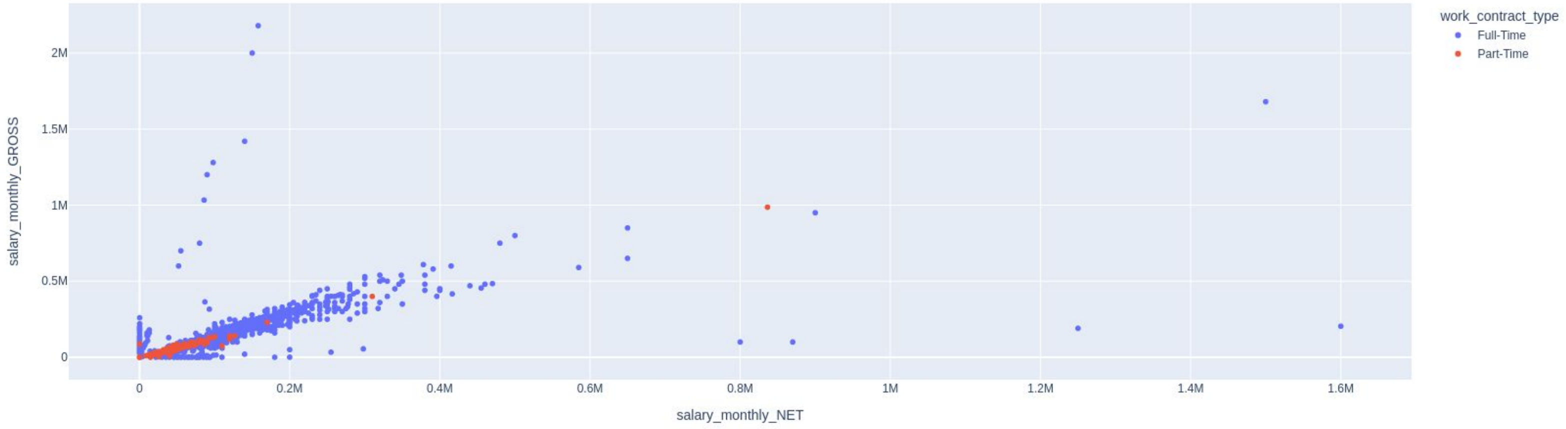
The catplots shows the **relationship between one categorical r.v.** X and **one numerical** Y.




```
sns.catplot(
data=df,
y='salary_monthly_NET',
x='salary_satisfaction'
)
```

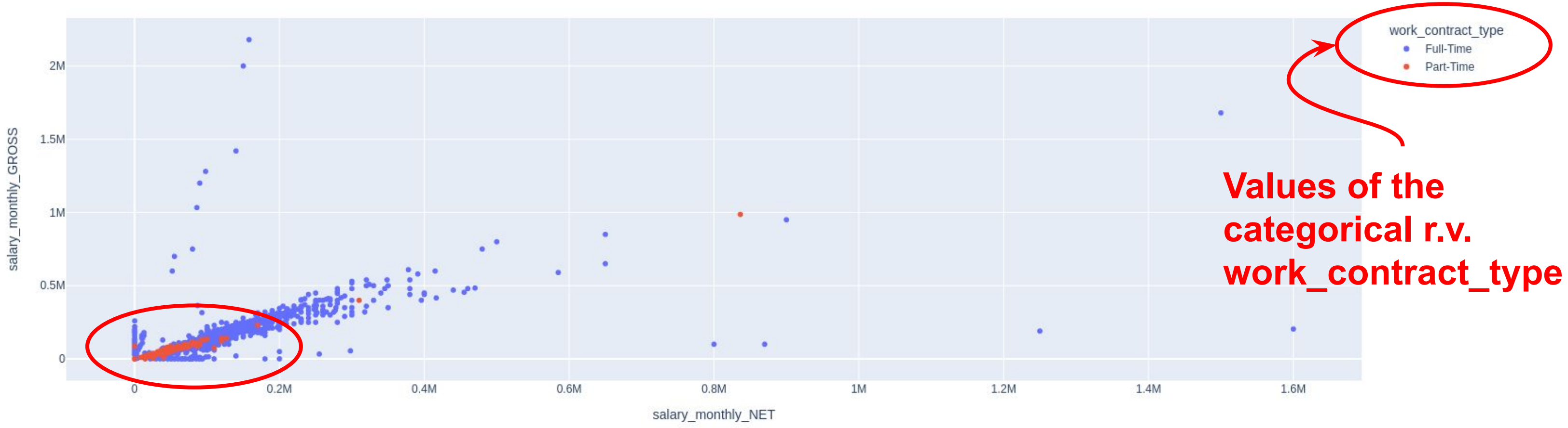
The power of hue

It controls when you want to **visualize different subsets of the data**. It can be used with **only categorical variables** and it is preferable where the **number of categories is small**.



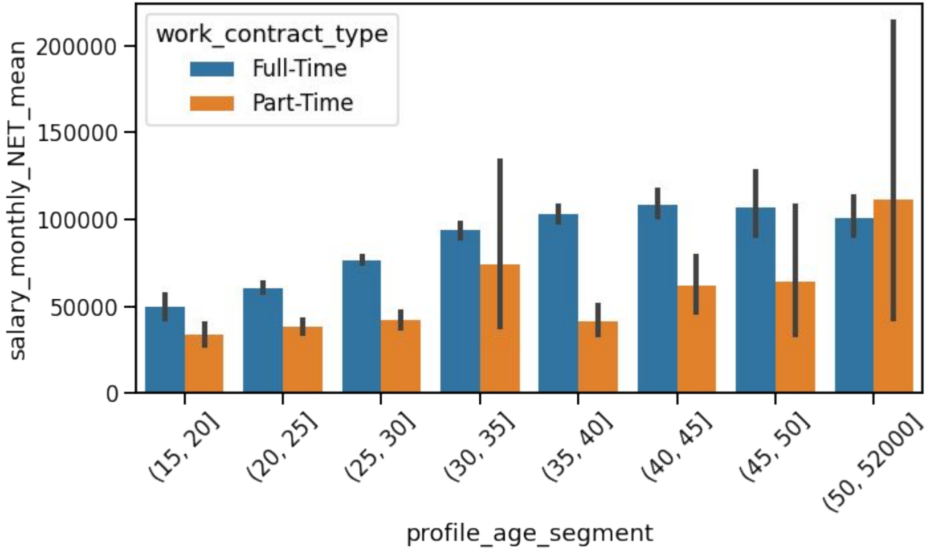
The power of hue

It controls when you want to **visualize different subsets of the data**. It can be used with **only categorical variables** and it is preferable where the **number of categories is small**.



The power of hue

It controls when you want to **visualize different subsets of the data**. It can be used with **only categorical variables** and it is preferable where the **number of categories is small**.

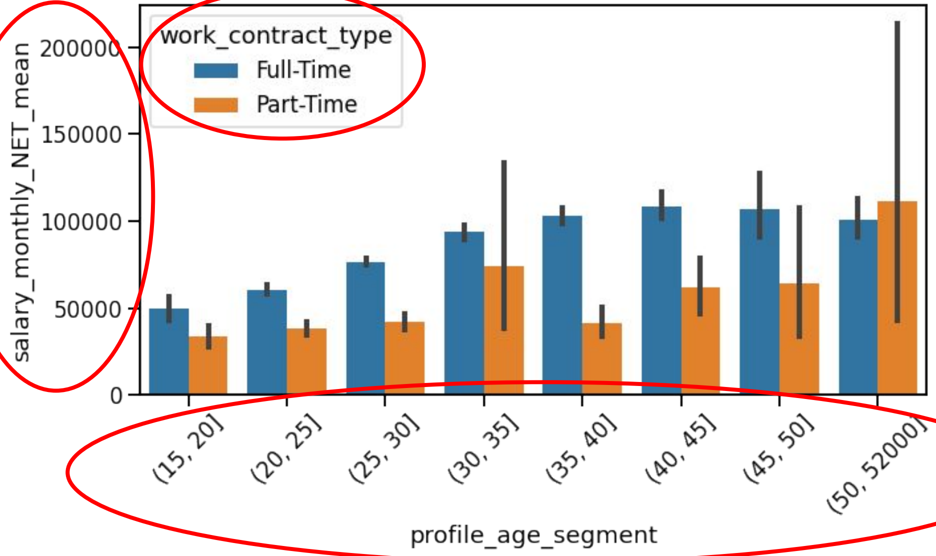


The power of hue

It controls when you want to **visualize different subsets of the data**. It can be used with **only categorical variables** and it is preferable where the **number of categories is small**.

(categorical r.v.)

(numerical r.v.)



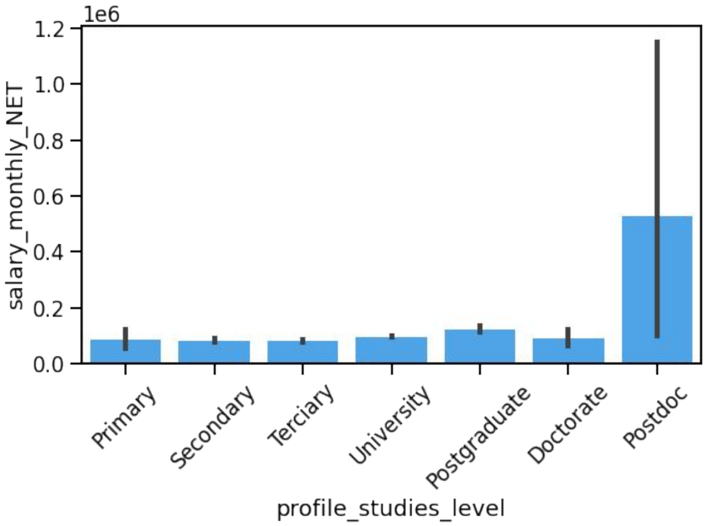
(categorical r.v.)

Demo with notebook

03_visualizing_relationships_of_rv.ipynb

From Seaborn to Plotly

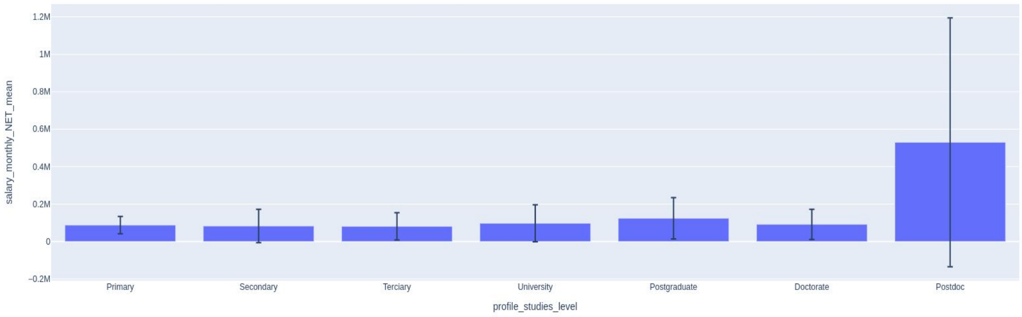
The plotly library is an interactive open-source plotting library that supports the creation of **more personalized plots than seaborn** but at the same time with a harder learning curve.




```
seaborn.barplot(
data=df,
y="salary_monthly_NET",
x='profile_studies_level',
estimator=numpy.mean,
ci=95)
```

From Seaborn to Plotly

The plotly library is an interactive open-source plotting library that supports the creation of **more personalized plots than seaborn** but at the same time with a harder learning curve.

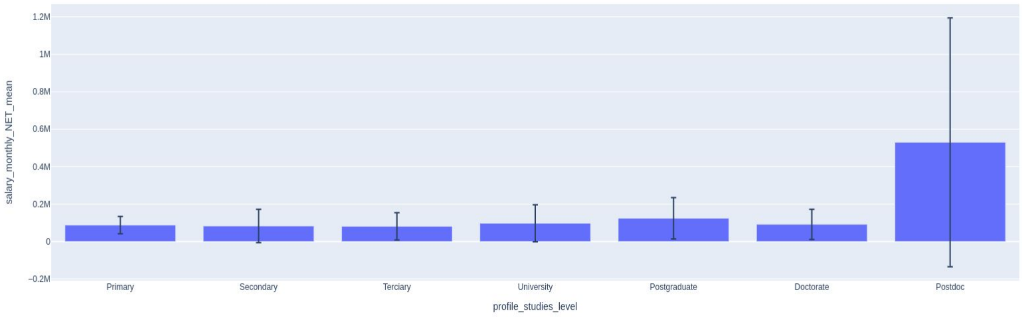



```
fig = px.bar(
df_studies_level_mean,
x='profile_studies_level',
y='salary_monthly_NET_mean',
error_y="salary_monthly_NET_std")
fig.show()
```

From Seaborn to Plotly

The plotly library is an interactive open-source plotting library that supports the creation of **more personalized plots than seaborn** but at the same time with a harder learning curve.

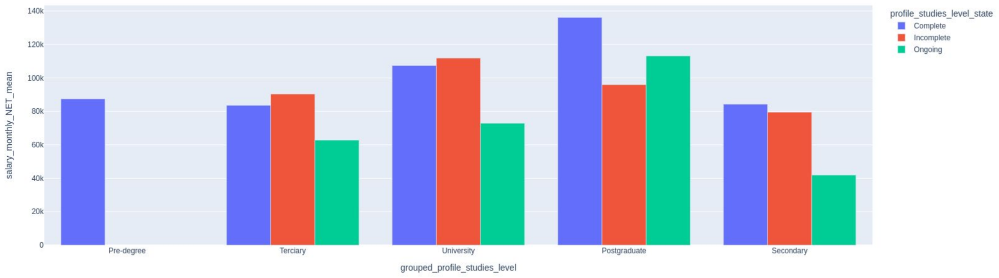
IMPORTANT! Sometimes we need to calculate the aggregation




```
fig = px.bar(
df_studies_level_mean,
x='profile_studies_level',
y='salary_monthly_NET_mean',
error_y="salary_monthly_NET_std")
fig.show()
```

From Seaborn to Plotly

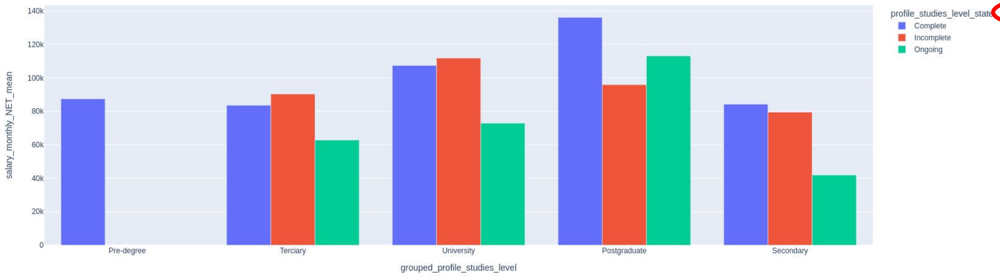
The plotly library is an interactive open-source plotting library that supports the creation of **more personalized plots than seaborn** but at the same time with a harder learning curve.




```
fig = px.bar(
df_grouped_studies_level_mean,
x='profile_studies_level',
y='salary_monthly_NET_mean',
color='profile_studies_level_state',
barmode='group')
fig.show()
```

From Seaborn to Plotly

The plotly library is an interactive open-source plotting library that supports the creation of **more personalized plots than seaborn** but at the same time with a harder learning curve.



Dataframe with the studies level, level state, and salary mean

```
fig = px.bar(
df_grouped_studies_level_mean,
x='profile_studies_level',
y='salary_monthly_NET_mean',
color='profile_studies_level_state',
barmode='group')
fig.show()
```

Demo with notebook
04_plotly_vs_seaborn.ipynb