Visualizing pandas DataFrames

Eric_A_Moore@yahoo.com

https://github.com/eamoore

8/18

“The Problem”: visual DataFrame dump

See what’s in a DataFrame, toward identifying interesting relationships among variables

- Concisely
- Clearly
- Don’t overwhelm
- Don’t overlook what matters

Big picture goal: making statistical data analysis easier to understand and do.

Approach

Code with: pandas, Altair, Vega Lite

Altair alternatives: MatplotLib, Seaborn, Bokeh, plotly

Other systems: Trifacta, Wrangler (UW IDL), Empirical

Other “dump” projects: focus on data vs variables, stats

Initial application: how much do athlete's bodies predict their position or sport? (baseball, football, Olympics)

Key Ideas

Simplify – analyses into phases, single purpose views

Simple schema – permute over to generate scenes and views

Simple invocation – developer effort matters much

Embody the visual judgment – best chart for the job

Two usages:

- **Explore**: Interactive, screen first – sequence, scroll, page, gesture
- **Explain**: Legible static image, highlight the “tell”

Phases of Analyses

1. ***Data***: Quality, Quantity, Qualify.
   - **Fields**: Formats? Gaps, causes for gaps? Need for transformation?
2. ***Variables***: What is potentially informative – similar, different, distinct?
   - See **Variables** – distribution of values
   - See **Variable – Variable** relationships
3. ***Model***: what is predictive?; what are useful abstractions?
   - See relationships in context of a Target Variable
   - Measures on Measures – toward statistical inference. **Category - Distribution**
   - "Fit" or abstract variable distributions and relationships. **Variable - Distribution**
   - "Focus" a predictive model to most essential form. Variable Graph.
   - "Test" competing hypotheses – **Distribution - Distribution**

Embody visual judgment

Describe, Compare … Fields, **Variables**, Distributions

Subject	Chart	Comments
Categorical 1D	Horizontal Histogram	Group by category.
Scalar 1D	Vertical Histogram	Binned X. Fit later w/ parametric Distribution
Category vs Category	Crosstab Table	Counts
Measure vs Measure	Scatter Plot	Fit later w/ Loess or regression.
Measure over Category	Hozo Bar Chart	Measure, conditioned on each category.
Measures on Measure over Category	Hozo Box Chart	Using mean, std deviation vs median and quartiles. “Separation” in 1D, with one measure.
Category over Measures	Scatter Plot with Category	Separation in 2D, with two joint measures. Only compare 2 or 3 categories at once.
Category over Category differences	Crosstab Table with residual density	Highlight non uniformity in joint distribution
Set of Correlations	Hozo Bar Chart (Correlations)	Highlight strong or weak correlations

Simple Schema

Statistics domain

- Compare effects of a process across subpopulations
- Effects are measured, estimated given samples of a population
- *Category* – divides population
- *Measure* -- describes population
- Distributions – key operand. Product of measures over measures. Tests between.

Data domain

- Symbolic -- *Categorical, Nominal, Ordinal, Key*
- Numeric -- *Quantitative, Temporal*
- *Set* ~ Sample ~ Population – collection of records
- Field ~ Variable ~ Feature
- Distributions are “features” of aggregates

Simple Schema

```
view_phases = [ 'Data', 'Variables', 'Model' ]

view_schema = {
    'Data' : [] ,
    'Variables' : [ '1D', '2D' ],
    'Model' : [ 'CategoryCategory', 'CategoryMeasure', 'Correlations' ] }

schema = {
    'Collection' : 'Baseball',
    'Measures' : [ 'Age', 'BMI', 'Height', 'Weight' ],
    'Categories' : ['Position' , 'Team'],
    'Target' : [ 'Position' ]
}
```

Simple invocation

```
schema['Sport'] = 'Baseball'
df_baseball = load_baseball('./baseball2016.csv')
df_baseball_summary = make_summary_df(df_baseball, schema)
view_all(df_baseball, df_baseball_summary, schema, phases, view_schema)
```

```
def load_baseball(file):
    df = pd.read_csv(file)
    # format df.
    baseball_columns_map = {
        'Position': 'Position',
        'Height(inches)': 'Height',
        'Weight(pounds)': 'Weight',
        'Age' : 'Age'
    }

    df = df.drop('Name', axis=1)
    df.rename(columns=baseball_columns_map,inplace=True)

    df['BMI'] = calculate_BMI(df['Height'], df['Weight'])
    return df
```

Pandas load.

Map data fields to schema names.

Calc joint and conditional distributions (summary_df).

Generate views.

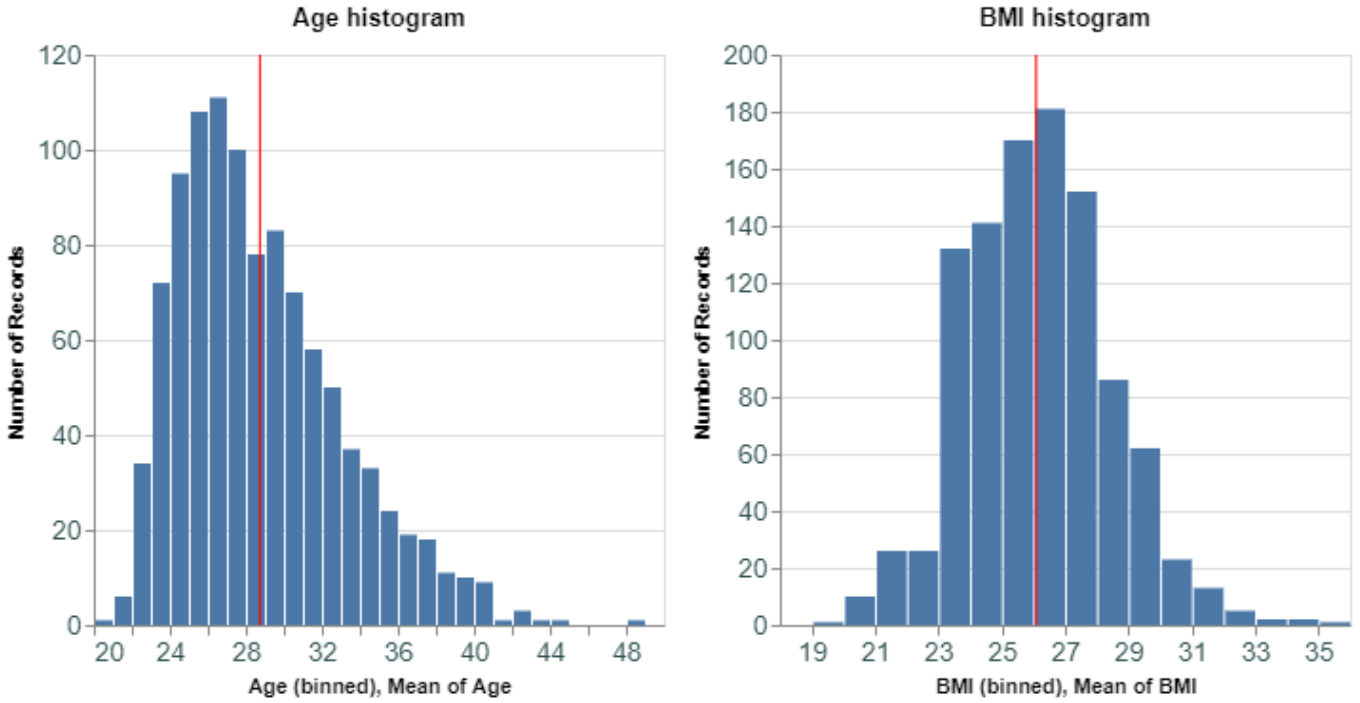
Charts in Altair

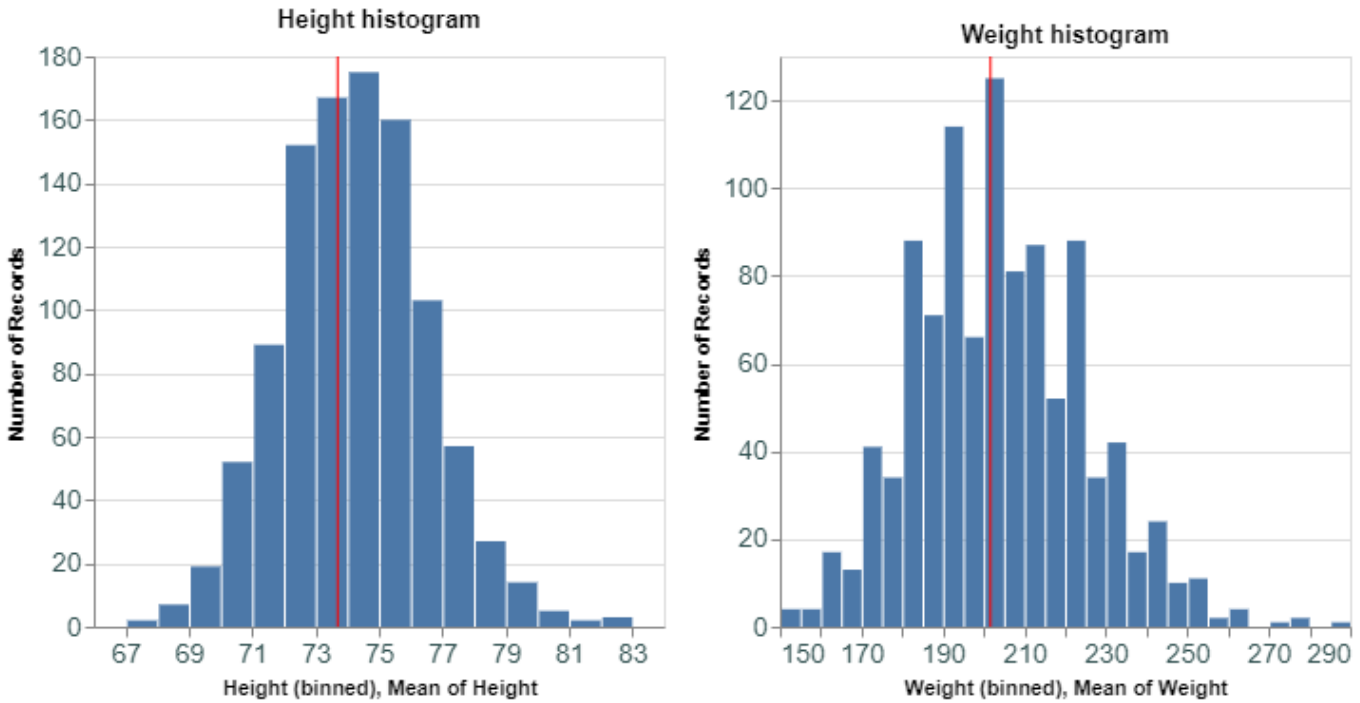
(DataFrame + field) + (axis + encoding + mark) = chart

- Vega, Vega Lite from DataViz Visionaries @UW IDL, Altair from Jake VanderPlas.
- Altair in python emits JSON spec to Vega Lite, Vega, to synthesize and render.
- Tidy data preferred
- Composition of chart by layers
- Peered charts. Shared Axes.
- Easy to specify interactive and coordinatedelements (Vega Lite)
- Data transformation – binning, sort, grouping

Challenges (for me)

- Composition of scenes (working outside Notebooks)
- Navigation events handled outside the chart (how to “click through”) **
- Resolution of conflicting attributes in scope hierarchy – axes, domains
- Should know how Vega Lite, Vega work





Histogram

Measure.

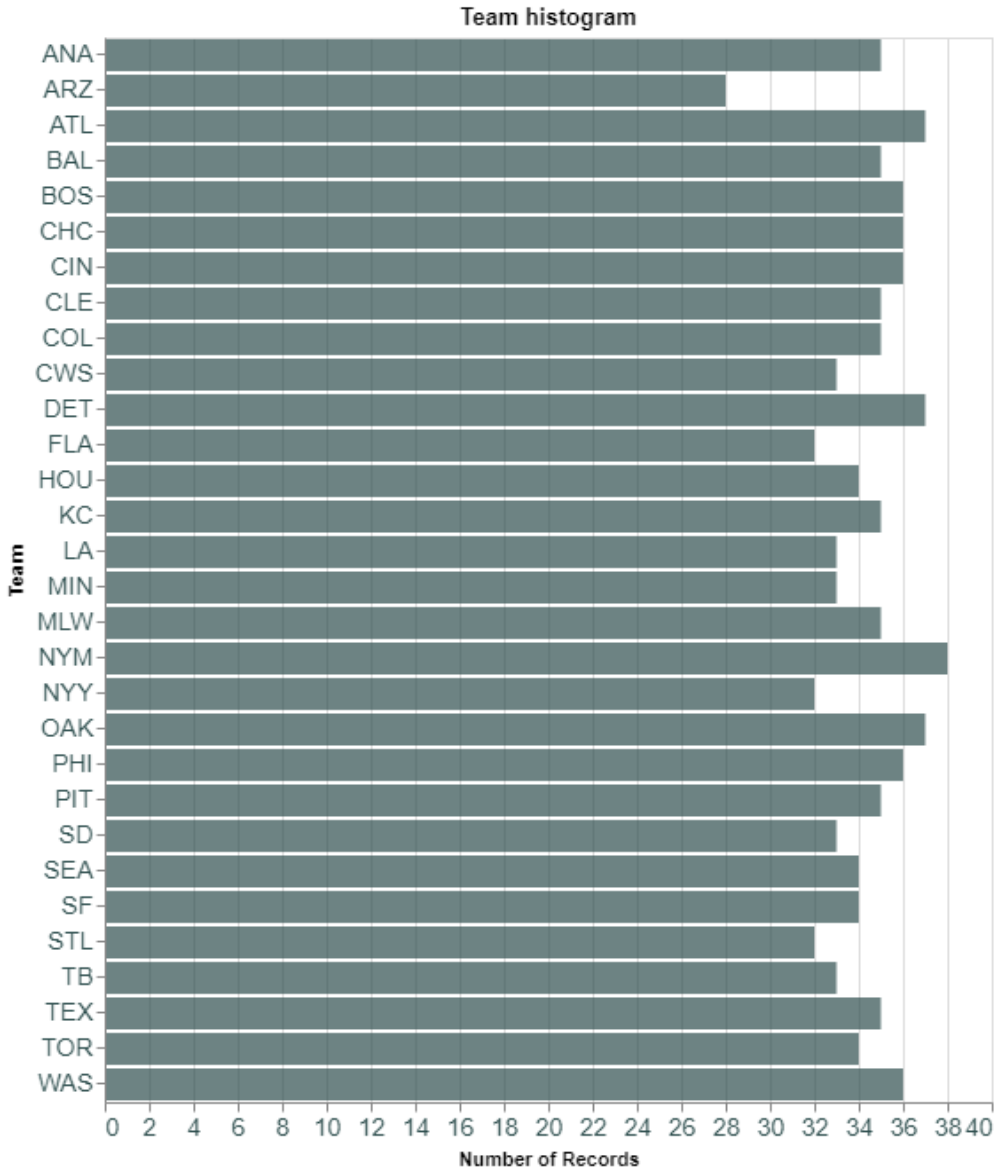
Altair’s binning. Bin upstream of viz …

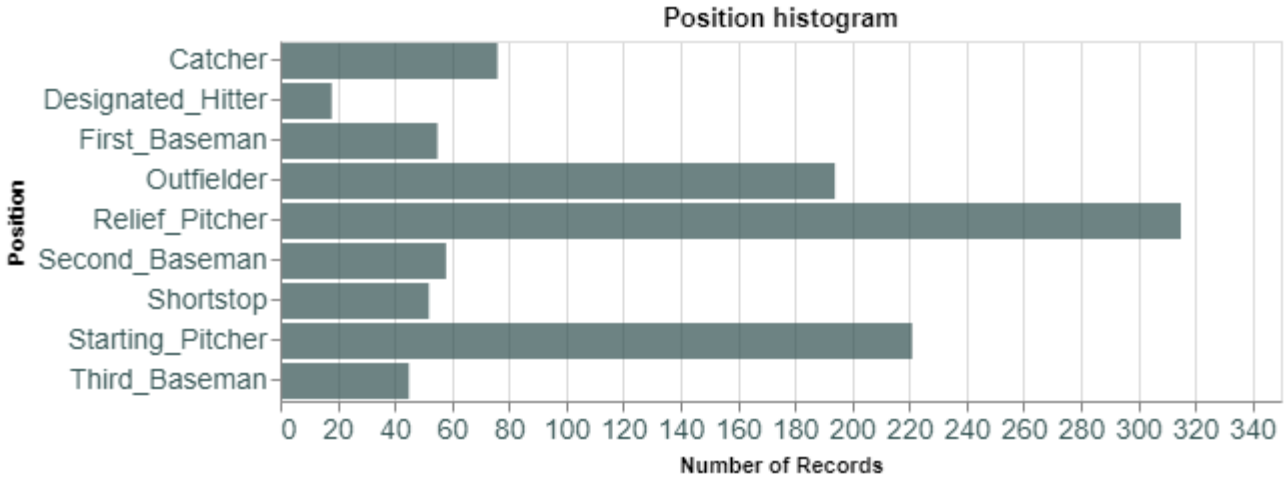
Nice to have: Slider for different bin sizes.

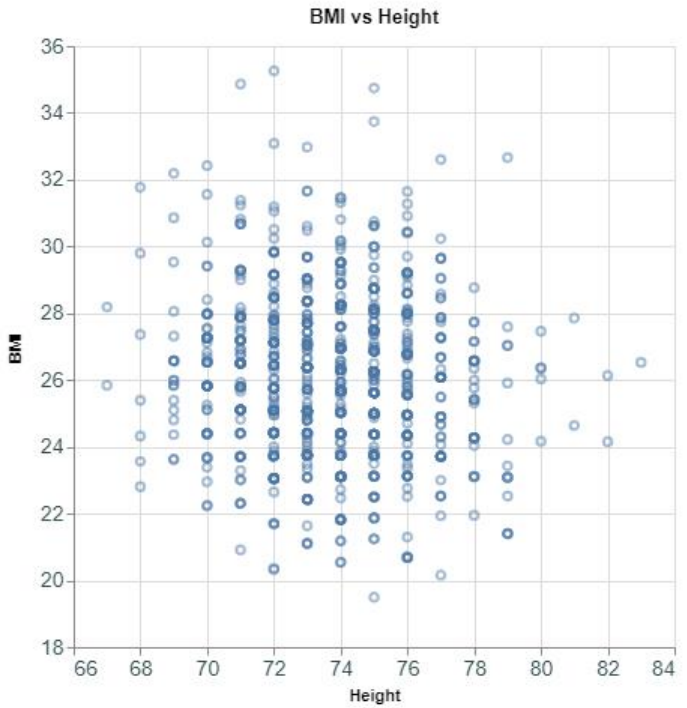
Hozo Histogram

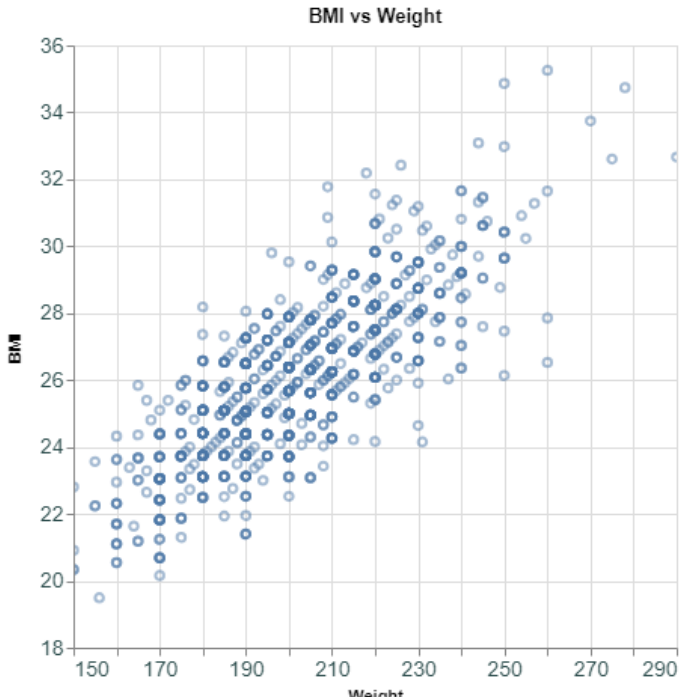
Category var.

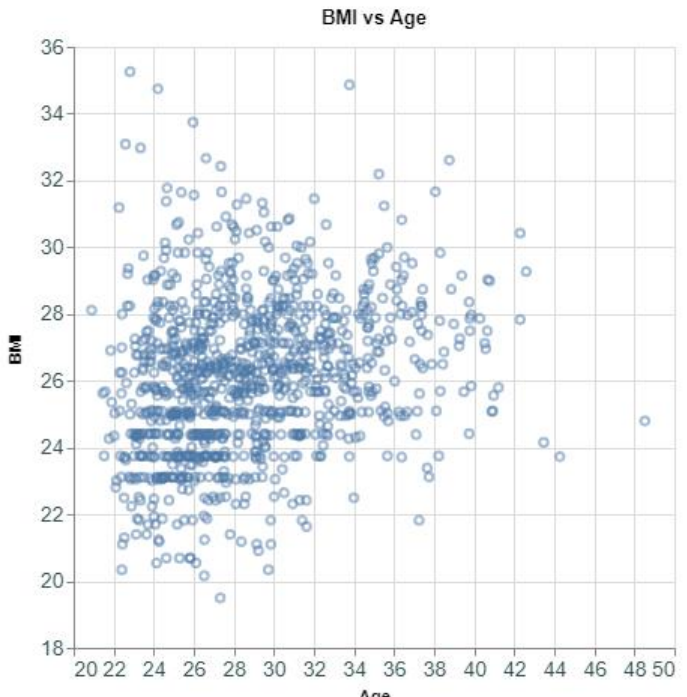
Bin by Category, sort by category or measure.
Nice to see counts.







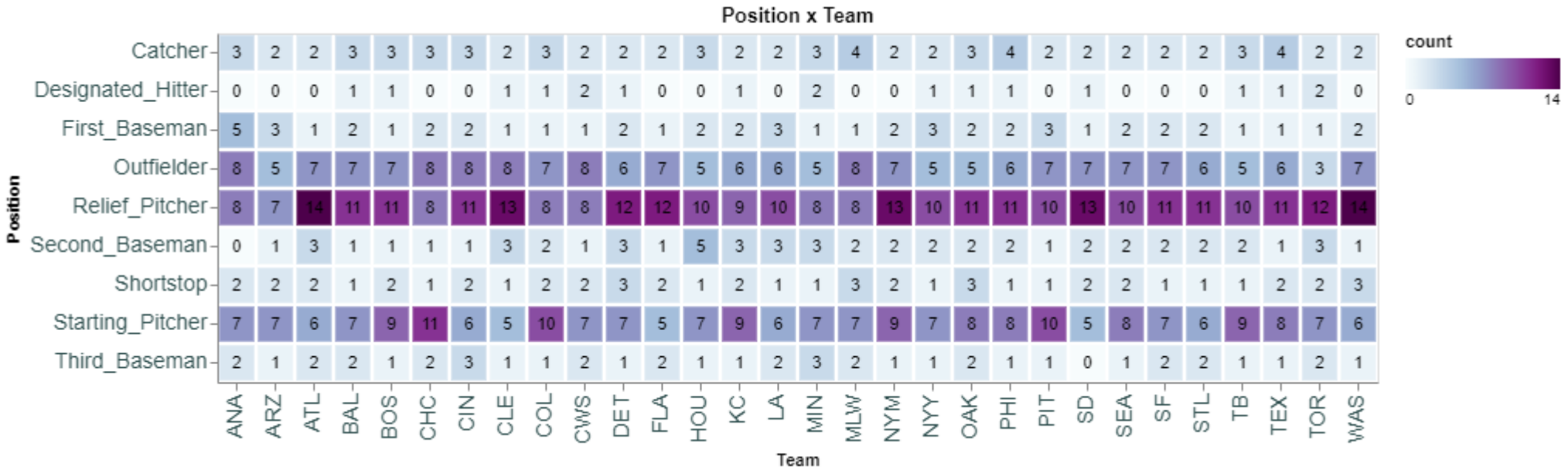




Scatter Plot

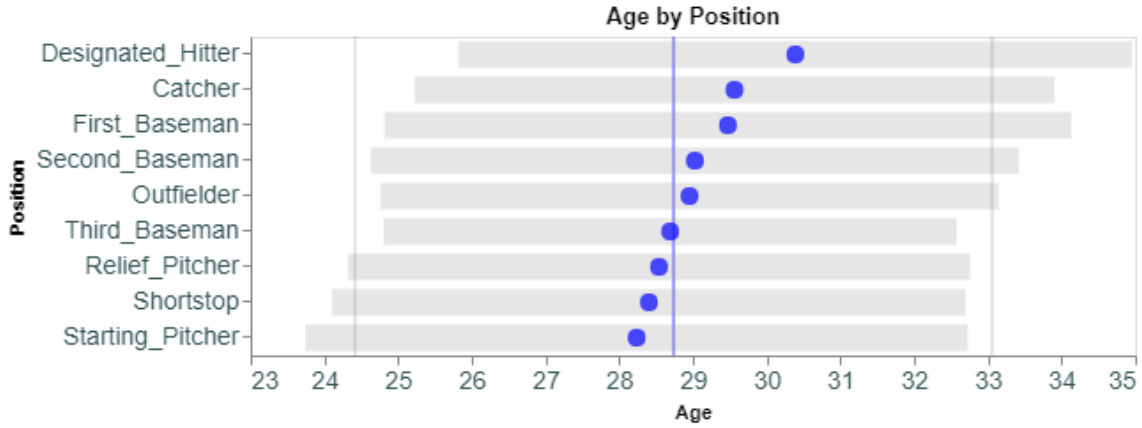
Altair/Vega Lite nicely interactive. Select datums or areas, pan/zoom.

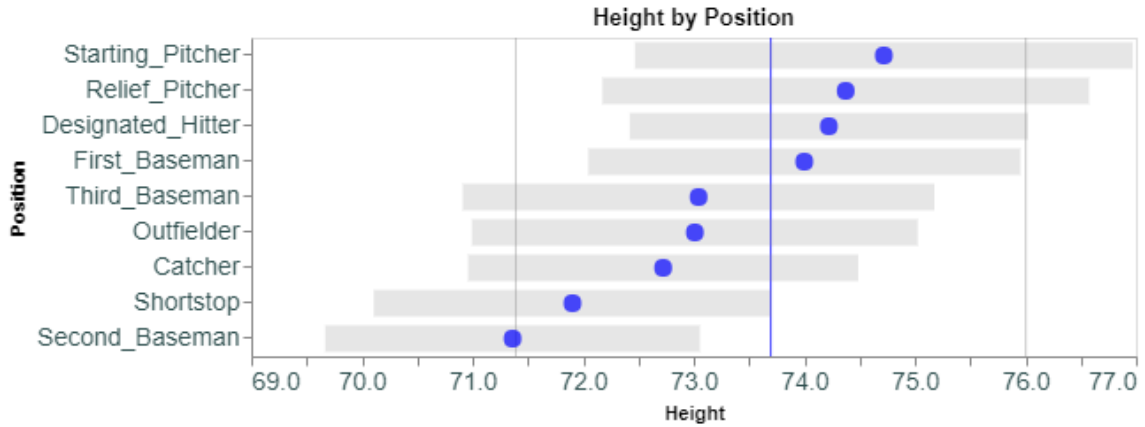
Cross Tab – Category vs Category

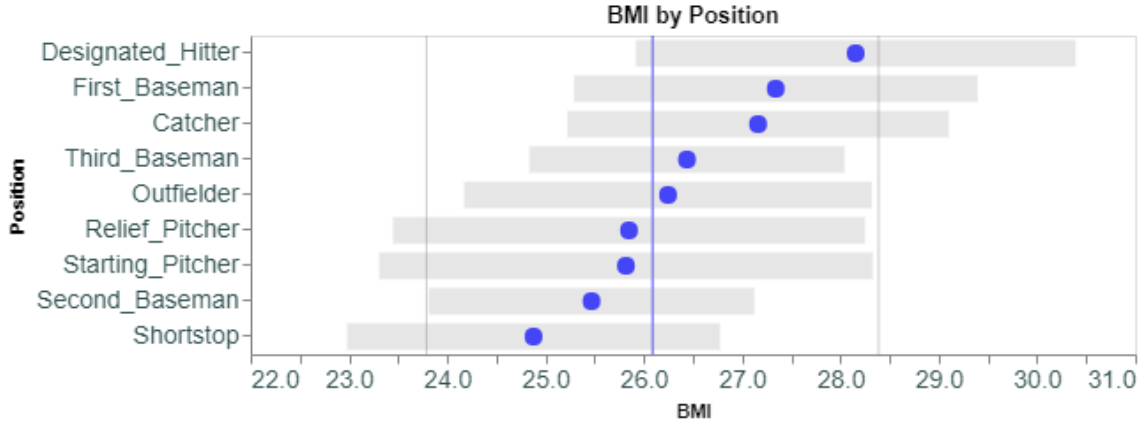


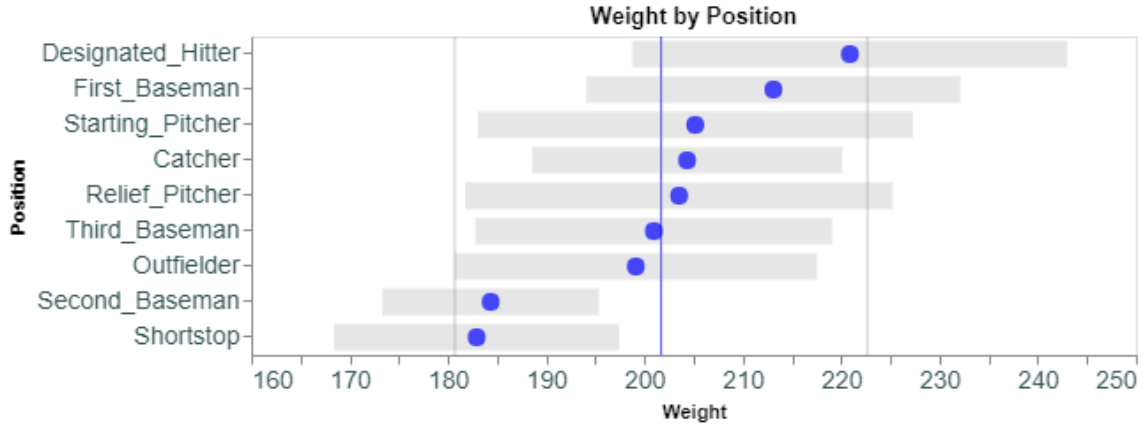
Counts. Compare variations across teams? *Normalize.*

“Box” Chart. Category vs Distribution of Measure.





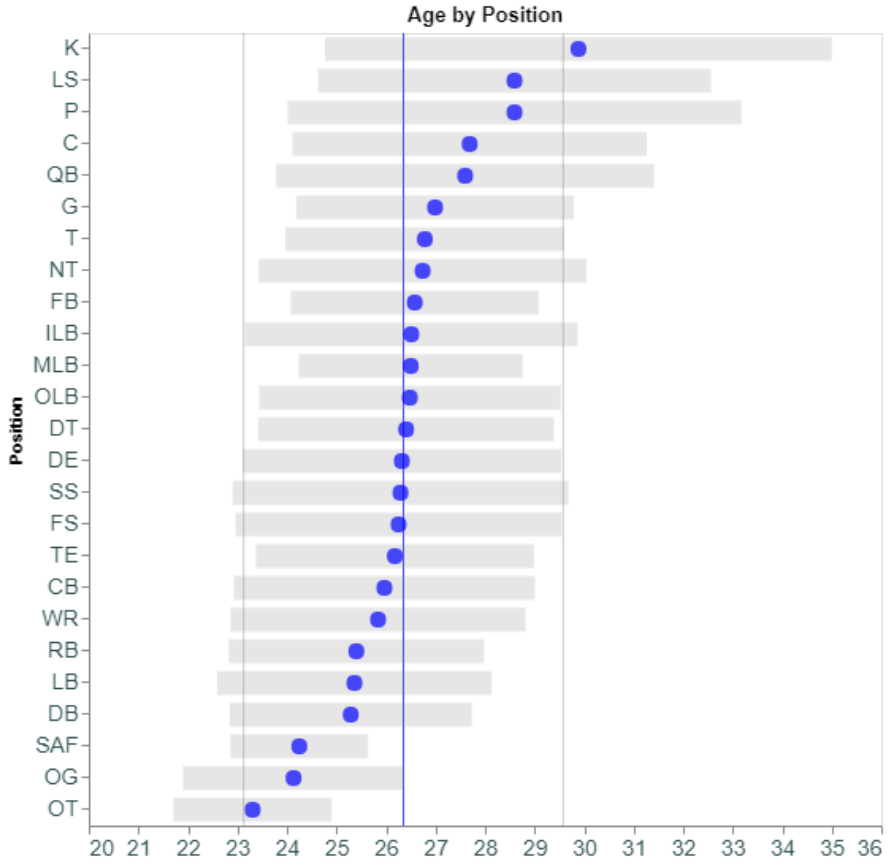


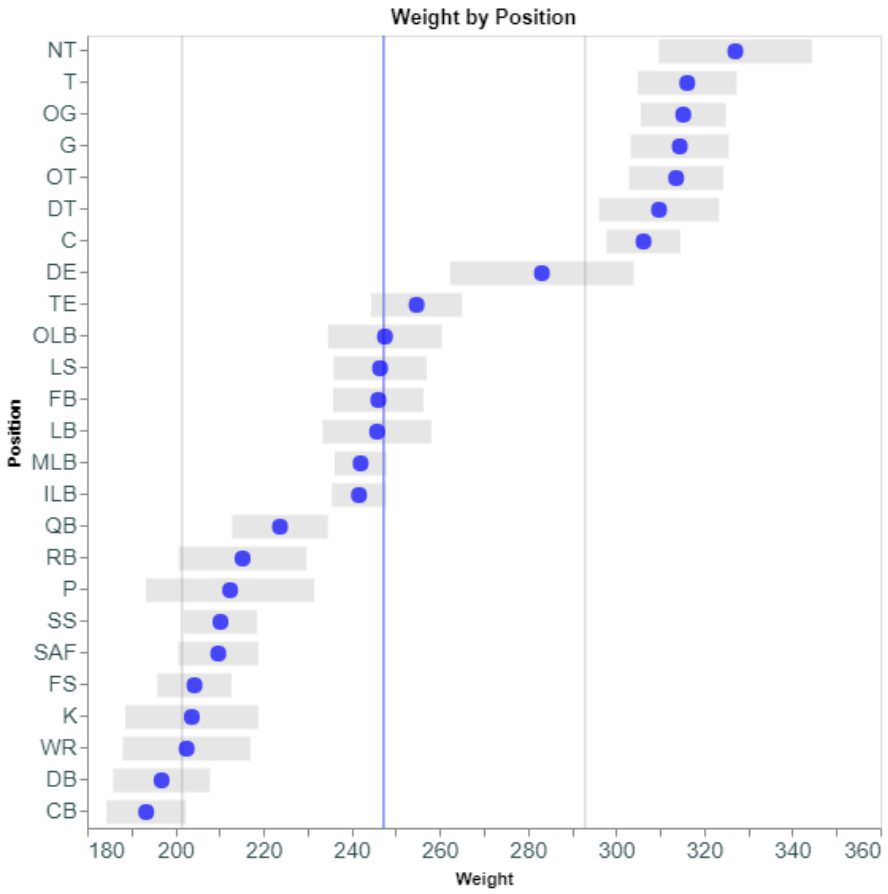


Classic Box: median, quartiles, max/min whiskers.

This: mean, std deviation. Rules – population mean, std. Rotated: better comparison.

Football … “Box”.

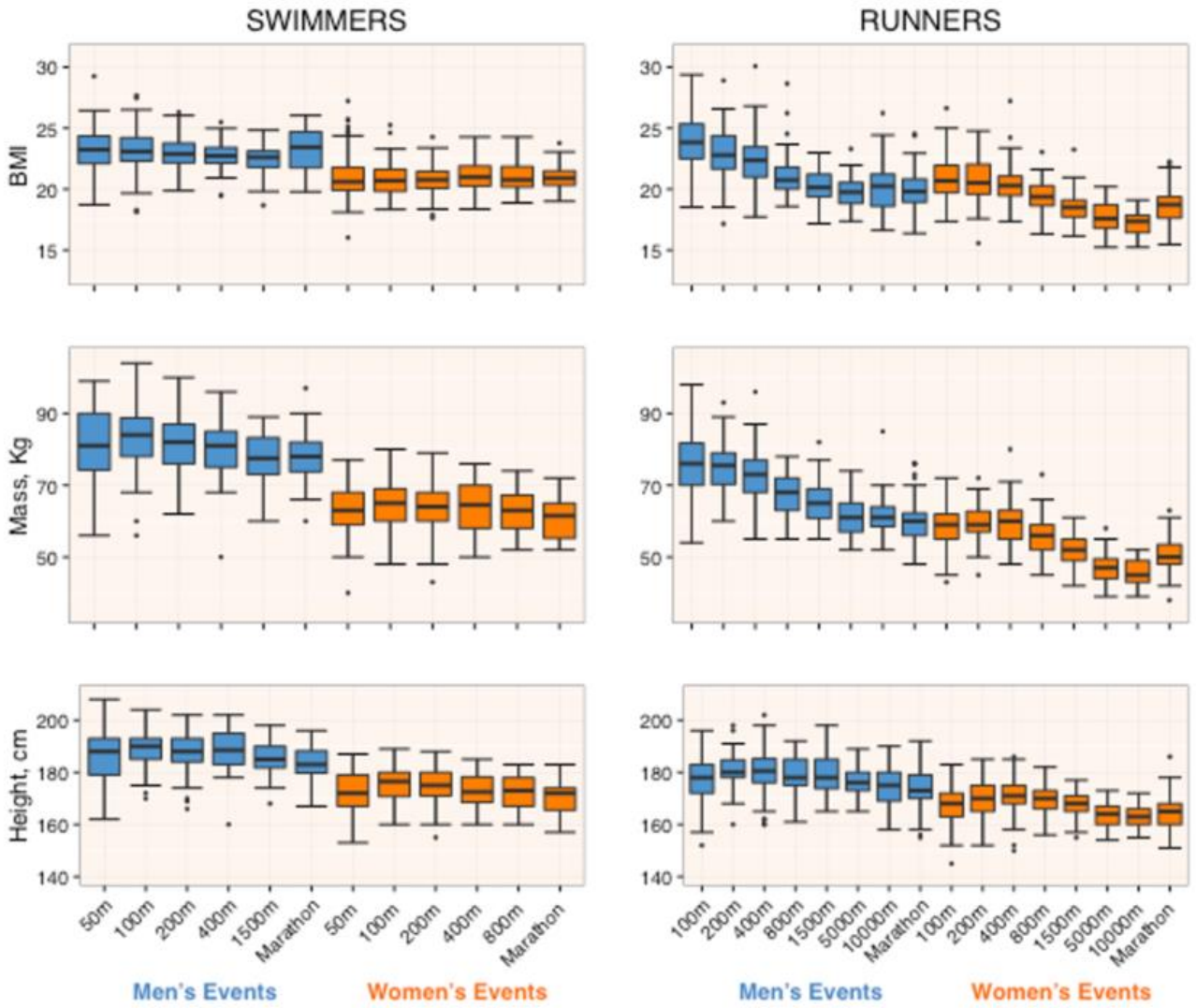




Visually, clear separation by weight for Position. Baseball too.

What’s next? How then do I tell of if the space is statistically meaningful?

Swimmers vs Runners, Gagnon, et al (2018)

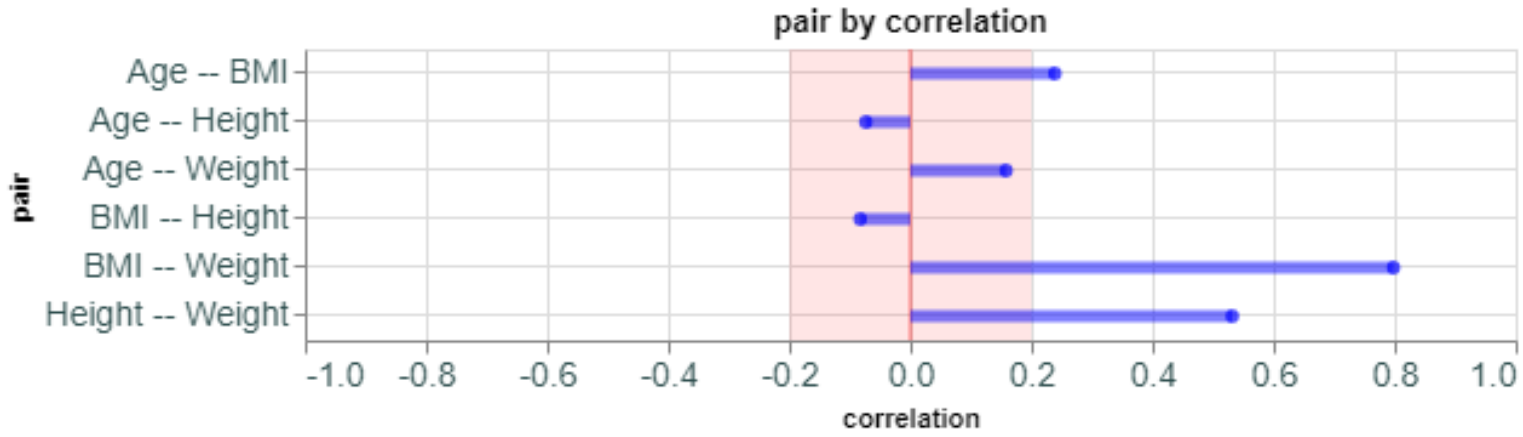


Classic Box Charts. Exhibit of evidence.

Visual story supports their analysis. Note consistent Y scales for juxtaposed charts.

Essential story is that BMI's don't change much for swimmers over event distance, and for runners they do.

Correlations



Correlations are not enough by themselves, without context of variable “dependency”.

```python
def calculate_BMI(h, w):
    # inches => meters, pounds => kg
    w = w / 2.205
    h = h * 0.0254
    bmi = w / (h * h)
    return bmi
```

Next Steps

- Navigable views under Notebook
- Views:
  - Additional Variable visualizations (temporal → Series)
  - Viz for Data Questions
  - Statistical viz – distribution, regression, Loess “layers”; residual focused visuals
- Package as a learner’s library
- Measures for ranking views
- Context of inferential statistics workflow

References

Jake VanderPlas @PyCon2018 Exploratory Data Visualization with Vega, Vega-Lite, and Altair

altair-viz.github.io

github.com/jakevdp/altair-examples

vallandingham.me/altair_intro.html

University of Washington Interactive Data Lab https://idl.cs.washington.edu/

"Making Data Visual A Practical Guide to Using Visualization for Insight" -- Miriah Meyer, Danyel Fisher – Explains rubric of subjects, questions, and appropriate visualization

How Body Type May Determine Runners' and Swimmers' Destinies – NYTimes 08-14-18

Extras

DataViz design issues

What am I looking at?

→ Clearly display clear labels for fields

See *all* the data?

→ See foremost just what's important to see

Is this "a thing"? (is there an 'effect' here?)

→ Visualize statistical measures too

Scale # fields

→ Groups, hierarchies in schema

→ Rank and trim display via measures over measures

Scale # records

→Measure, aggregate, bin *before* handing off to display substrate

→ Work with aggregates and features of aggregates asap, sample before

Analyses – humans in the loop

To Judge –

to test a hypothesis,
to ask and answer a question in some medium

Information operations toward judgment

Describe, Compare, Abstract, Infer.

Visual “medium”:

Person, through DataViz.
Look → Judge

Computational “medium”:

System, through Algorithms.
Compute → Look → Judge.
Still need to see *something*.

Development Observations

Libraries … as collective works, and as works in progress

- Diffuse channels: docs, git issues (open & closed), project google group/slack, project contributor blogs, user blogs, contributor presentations
- Different ways to do the same thing … bushy interfaces

Library as a lever, vs utility

- Great to copy from Examples gallery
- “Why did that break?” Value in knowing what it really does, and how.