TamStat 2019
Technical Update

Michael Baas
What’s new?

- DUI
- API
- FontAwesome
- Misc
DUI

- Dyalog User Interface
- Project title only...
- Cross platform HTML5-based front ends
- DUI is...
DUI

- Dyalog User Interface
- Project title only...
- Cross platform HTML5-based front ends
- DUI is...

Goethe
"Two souls alas! are dwelling in my breast!"
DUI

- Dyalog User Interface
- Project title only...
- Cross platform HTML5-based front ends

DUI is...

- Content Generation (HTML, JS, CSS)
- Server (serves content)
- HRServer
- MiServer

Goethe
"Two souls alas! are dwelling in my breast!"
Transition to DUI

- When I got it on my PC, I launched it...
- ...started TamStat...
- ...and it worked!
- Almost...
Transition to DUI

- When I got it on my PC, I launched it...
- ...started TamStat...
- ...and it worked!
- Almost....
- There was only one thing that needed to change
Overriding functions?

If you used the overrides of methods exposed by the Server-Class (onServerLoad, onServerStart, onSessionStart, onSessionEnd, onHandleMSP, onHandleRequest, onIdle, Error, Log, Cleanup), you previously defined these in your custom Server-Class:
Overriding functions?

- If you used the overrides of methods exposed by the Server-Class (onServerLoad, onServerStart, onSessionStart, onSessionEnd, onHandleMSP, onHandleRequest, onIdle, Error, Log, Cleanup), you previously defined these in your custom Server-Class
- Move them into ./Code/Overrides.dyalog
- ✔️ Done
It just works

- `]load {path}/dui[.dyalog]

or

- `)load {path}/dui

and then

- `#.DUI.Start'../TamStat-App' ʰ HTMLRenderer
- `#.DUI.Start'../TamStat-App' 8080 ʰ MiServer/Browser
API-Changes

- Last year: Namespaces as function arguments
API-Changes

- Last year: Namespaces as function arguments
- We made progress: now we're passing a namespace that contains a variable which is an **array of namespaces**!
- "Stupid" front end. But stupid in a very smart way.
Stupidly smart front end?

- The front end for HypoWiz Wizard knows almost nothing Hypothesis testing – only specific code deal with specific display of the result (using Thumbs-up/down)
- Front end is build around the "Grid" – an array of namespaces that defines the screen-elements (returned by the API)
Smartly stupid front end...in practice

Expression: 0.8 0.9 0.95 0.99 mean confInt show (Height)

ConfLevel: 80 %, 90 %, 95 %, 99 %

Parameter: mean

Operator: confInt

Sample Size: 38

Mean: 68.77632

Std Dev: 4.47621

Confidence Interval and Point Estimate graph
Smartly stupid front end...in practice

Expression: 0.8 0.9 0.95 0.99 mean confInt show (Height)

<table>
<thead>
<tr>
<th>ConfLevel</th>
<th>80 %</th>
<th>90 %</th>
<th>95 %</th>
<th>99 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>mean</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>confInt</td>
<td>Sample Size</td>
<td>Mean</td>
<td>Std Dev</td>
</tr>
<tr>
<td>Sample</td>
<td>Height</td>
<td>38</td>
<td>68.77632</td>
<td>4.47621</td>
</tr>
</tbody>
</table>

Bar chart showing confidence intervals for different confidence levels.
Smartly stupid front end...in practice

Expression: 0.8 0.9 0.95 0.99 mean confInt show (Height)

ConfLevel: 80 % 90 % 95 % 99 %

Parameter: mean

Operator: confInt

Sample: Height 38 68.77632 4.47621

Graph showing confidence intervals and point estimate.
# Smartly stupid front end...in practice

<table>
<thead>
<tr>
<th>ConLevel</th>
<th>80</th>
<th>90</th>
<th>95</th>
<th>99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>mean</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>confInt</td>
<td>Sample Size</td>
<td>Mean</td>
<td>Std Dev</td>
</tr>
<tr>
<td>Sample</td>
<td>Height</td>
<td>38</td>
<td>68.77632</td>
<td>4.47621</td>
</tr>
</tbody>
</table>

![Confidence Interval](image)
Smartly stupid front end...in practice
Smartly stupid front end...in practice
Smartly stupid front end...in practice
```
= ConfLevel
{"Type":"T"}
```

<table>
<thead>
<tr>
<th>ConfLevel</th>
<th>80 %</th>
<th>90 %</th>
<th>95 %</th>
<th>99 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>mean</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>confint</td>
<td>Sample Size</td>
<td>Mean</td>
<td>Std Dev</td>
</tr>
<tr>
<td>Sample</td>
<td>Height</td>
<td>38</td>
<td>68.77632</td>
<td>4.47621</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Confidence Interval

- 80%: (67.828, 69.724)
- 90%: (67.551, 70.608)
- 95%: (67.305, 70.248)
- 99%: (65.805, 70.748)

Confidence Interval: **Green**

Point Estimate: **Blue**
Smartly stupid front end...in practice
Parameter = mean

{"Items": ["mean", "proportion", "var", "sdev"], "Type": "d"}
Parameter = mean

{ "Items": [ "mean", "proportion", "var", "sdev" ], "Type": "d" }
Parameter = mean

{"Items": ["mean", "proportion", "var", "sdev"], "Type": "d"}
Parameter = mean

{"Items": ["mean", "proportion", "var", "sdev"], "Type": "d"}

<table>
<thead>
<tr>
<th>ConflLevel</th>
<th>Sample Size</th>
<th>Height</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>38</td>
<td>68.77632</td>
</tr>
</tbody>
</table>

Confidence Interval: 80% (67.629, 69.724)
90% (67.551, 70.601)
95% (67.305, 70.248)
99% (65.805, 70.748)
Sample = Height

{"Items" : "F", "Type" : "E"}
Sample = Height

{"Items":"F","Type":"E"}
Sample = Height

{ "Items": "F", "Type": "E" }

<table>
<thead>
<tr>
<th>Expression</th>
<th>0.99 95% 99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfLevel</td>
<td>80 %</td>
</tr>
<tr>
<td>Parameter</td>
<td>mean</td>
</tr>
<tr>
<td>Operator</td>
<td>confint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>Height</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>68.77632</td>
<td>4.47621</td>
</tr>
</tbody>
</table>

Confidence Interval | Point Estimate
--- | ---
90% (67.829, 69.724) | 68.77632
90% (67.531, 70.601) | 68.77632
95% (67.305, 70.288) | 68.77632
99% (65.805, 70.748) | 68.77632
Sample = Height

\{ "Items": "F", "Type": "E" \}

<table>
<thead>
<tr>
<th>ConfLevel</th>
<th>80</th>
<th>95</th>
<th>99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>mean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>confint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample</td>
<td>Height</td>
<td>38</td>
<td>68.77632</td>
</tr>
</tbody>
</table>

80% (67.829,69.724)
60% (67.551,70.464)
50% (67.373,70.286)
99% (65.805,70.748)

Confidence Interval Point Estimate
LbSample2 = --

{"Items": ["--", "Sample2", "splitBy"], "Type": "d"}
TamStat 2019 – Technical Update

LbSample2 = --

{"Items": ["--", "Sample2", "splitBy"], "Type": "d"}

<table>
<thead>
<tr>
<th>Expression</th>
<th>0.9999999999999999</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ConILevel</th>
<th>80</th>
<th>95</th>
<th>99</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>mean</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Operator</th>
<th>Sample Size</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>Height</td>
<td>38</td>
<td>68.77632</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Confidence Interval</th>
<th>Point Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% (67.829, 69.724)</td>
<td></td>
</tr>
<tr>
<td>95% (67.551, 70.604)</td>
<td></td>
</tr>
<tr>
<td>99% (67.105, 70.248)</td>
<td></td>
</tr>
<tr>
<td>99% (65.625, 70.748)</td>
<td></td>
</tr>
</tbody>
</table>
TamStat 2019 – Technical Update

LbSample2 = --

{"Items": ["--", "- Sample2", "splitBy"], "Type": "d"}

<table>
<thead>
<tr>
<th>ConfLevel</th>
<th>80</th>
<th>95</th>
<th>%</th>
<th>99</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample</th>
<th>Sample Size</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>38</td>
<td>68.77632</td>
<td>4.47621</td>
</tr>
</tbody>
</table>

Confidence Interval vs Point Estimate
FontAwesome

- A versatile collection of icons
- Dyalog supported the development and was rewarded with a Dyalog-D icon!
FontAwesome

- A versatile collection of icons
- Dyalog supported the development and was rewarded with a Dyalog-D icon!
- Upgraded to use Version 5.10.2

Fontawesome.com
> 7,000 Icons
FontAwesome

- A versatile collection of icons
- Dyalog supported the development and was rewarded with a Dyalog-D icon!
- Upgraded to use Version 5.10.2
- TamStat now uses Duotone-Icons
FontAwesome

- A versatile collection of icons
- Dyalog supported the development and was rewarded with a Dyalog-D icon!
- Upgraded to use Version 5.10.2
- TamStat now uses Duotone-Icons
- MiServer / DUI-Users are entitled to use it!
  - Available on my.dyalog.com (soon)
  - $99 Licence might be required if you want to deploy it with an app - check their terms.
Misc[2]: download from a private repo!

Write a script to download a release from a private GitHub-Repository!

- curl (Windows)
Misc[2]: download from a private repo!

Write a script to download a release from a private GitHub-Repository!

- curl (Windows)
Misc[2]: download from a private repo!

Write a script to download a release from a private GitHub-Repository!

- curl (Windows)
Misc[2]: download from a private repo!
Misc[2]: download from a private repo!

Write a script to download a release from a private GitHub-Repository!

- **curl (Windows)**
- **HttpCommand**

```dylan
]load HttpCommand
hc←⎕NEW HttpCommand
hc.Headers←(('Authorization' 'token ***')
             ('Accept' 'application/vnd.github.v3.raw'))
hc.URL←'https://api.github.com/repos/Dyalog/' , repo , '/releases'
res←hc.Run
:If res.rc=0
    js←⎕JSON res.Data
    js←js[⍸5<≢¨js.zipball_url]  ⍝ filter & sort by published_at
    hc.URL←js[1].zipball_url
    res2←hc.Run
    ...res2.Data ⎕NAPPEND tie...
```

#dyalog19