

# pandoc2pdf test

me

## DataFrame

```
using DataFrames
using PrettyTables
using Latexify

df = DataFrame("a" => 1, "b" => 2, "c" => 3)
```

```
1×3 DataFrame
Row   a      b      c
      Int64 Int64 Int64

  1      1      2      3
```

## Latexify

```
latexify(df)
```

a	b	c
1	2	3

## PrettyTable

```
df |> pretty_table
```

```
      a      b      c
Int64 Int64 Int64

  1      2      3
```

## Default

```
pretty_table(df, tf=PrettyTables.tf_latex_default)
```

```
\begin{table}
  \begin{tabular}{rrr}
```

<b>a</b>	<b>b</b>	<b>c</b>
Int64	Int64	Int64
1	2	3

```

\hline\hline
\textbf{a} & \textbf{b} & \textbf{c} \\
\texttt{Int64} & \texttt{Int64} & \texttt{Int64} \\\hline
1 & 2 & 3 \\\hline\hline
\end{tabular}
\end{table}

```

## Simple

```
pretty_table(df, tf=PrettyTables.tf_latex_simple)
```

```

\begin{table}
\begin{tabular}{rrr}
\hline
\textbf{a} & \textbf{b} & \textbf{c} \\
\texttt{Int64} & \texttt{Int64} & \texttt{Int64} \\\hline
1 & 2 & 3 \\\hline
\end{tabular}
\end{table}

```

## Book Tabs

```
pretty_table(df, tf=PrettyTables.tf_latex_booktabs)
```

```

\begin{table}
\begin{tabular}{rrr}
\toprule
\textbf{a} & \textbf{b} & \textbf{c} \\
\texttt{Int64} & \texttt{Int64} & \texttt{Int64} \\\midrule
1 & 2 & 3 \\\bottomrule
\end{tabular}
\end{table}

```

## Alternative

Reproducing issue: Incorporating LaTeX output from code chunk.

## Latex Backend

```
pretty_table(df, backend = :latex)
```

## Markup Backend

```
pretty_table(df, backend = :markup)
```

## Function

Another alternative solution offered from issue: Visualizing a table in weave report?.

```
function latextable(df)
    io = IOBuffer();
    show(io, MIME("text/latex"), df; eltypes = false);
    print(String(take!(io)));
end
```

latextable (generic function with 1 method)

Test the function latextable.

```
latextable(df)
```

	a	b	c
1	1	2	3