tidyr

author: Etienne Low-Décarie date: September 24, 2015

Long vs wide data

Wide `{r fig.width=6, fig.height=3,echo=F,message=FALSE} require(tidyr) require(gridExtra) require(grid) grid.newpage() #clear the graphic device grid.table(head(iris)) #create a nice graphic table`

Long `{r fig.width=6, fig.height=6,echo=F} long_iris<-gather(iris,"Measurement","Value", -Species) grid.newpage() grid.table(head(long_iris,15))`

Tidy vs untidy data

Tidy data

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

Messy data - Anything else

Wickham, H. (2014). Tidy Data. J. Stat. Softw., 59, 1–2.

History

- reshape and reshape2 -melt and cast -aggregate: summary calculations
- tidyr -only data frames -simple unique use verbs -no summarising/aggregation

Going from wide to long

class: small-code

`gather`

(`melt` in reshape(2))

`{r fig.width=6, fig.height=6,eval=F} long_data<-gather(wide_data, key, value, selected_columns)`

Ways to select columns

- Use bare variable names. `{r fig.width=6, fig.height=6,echo=F} long_iris<-gather(iris,"Measurement", "Value", Sepal.Length, Sepal.Width, Petal.Length, Petal.Width)`

Ways to select columns

- Select all variables between x and z with x:z `{r fig.width=6, fig.height=6,echo=F} long_iris<-gather(iris,"Measurement", "Value", Sepal.Length:Petal.Width)`

Ways to select columns

- Exclude y with -y. `{r fig.width=6, fig.height=6,echo=F} long_iris<-gather(iris,"Measurement", "Value", -Species)`

Going from long to wide

`spread`

((d/a)cast in reshape(2))

`{r fig.width=6, fig.height=6,eval=F} wide_data <- spread(long_data, key, value)`

Going from long to wide

`{r fig.width=6, fig.height=6,echo=T, eval=F} wide_iris <- spread(long_iris, Measurement, Value)`

Going from long to wide

Each case must have a label!

“‘{r fig.width=6, fig.height=6,echo=T} iris$Specimen <- 1:nrow(iris)

long_iris<-gather(iris,“Measurement”, “Value”, Sepal.Length:Petal.Width)

wide_iris <- spread(long_iris, Measurement, Value) “‘

Going long for faceting by variable

Excellent for exploratory analysis

```
{r fig.width=6, fig.height=6,echo=T} require(ggplot2) p <- qplot(data=long_iris,
x=Species,            y=Value,            geom="bar",            stat="summary",
fun.y="mean",            fill=I("grey"))+   stat_summary(fun.data
= "mean_cl_boot", geom="errorbar")
```

```
{r fig.width=6, fig.height=6,echo=F} print(p)
```

Going long for faceting by variable

```
{r fig.width=6, fig.height=6,echo=T} print(p+facet_grid(.~Measurement))
```

Going long for faceting by variable

```
{r fig.width=3, fig.height=3,echo=T} print(p+facet_grid(Measurement~.,
scale="free"))
```

Seperate string variable

```
{r fig.width=3, fig.height=3,echo=T} seperated_iris <- separate(long_iris,
Measurement,                        c("Organ", "Dimension"))
```

Seperate string variable and spreading

```
{r fig.width=3, fig.height=3,echo=T} wide_iris <- spread(seperated_iris,
Dimension,                    Value)
```

Plot seperated iris

```
{r fig.width=6, fig.height=6,echo=T} p <- qplot(data=seperated_iris,
x=Species,            y=Value,            geom="bar",            stat="summary",
fun.y="mean",            fill=I("grey"))+   stat_summary(fun.data
= "mean_cl_boot",                geom="errorbar")+   facet_grid(Organ~Dimension,
scale="free")
```

```
{r fig.width=4, fig.height=5,echo=F} print(p)
```