Part 1: Concepts and Terminology

"It's hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation."

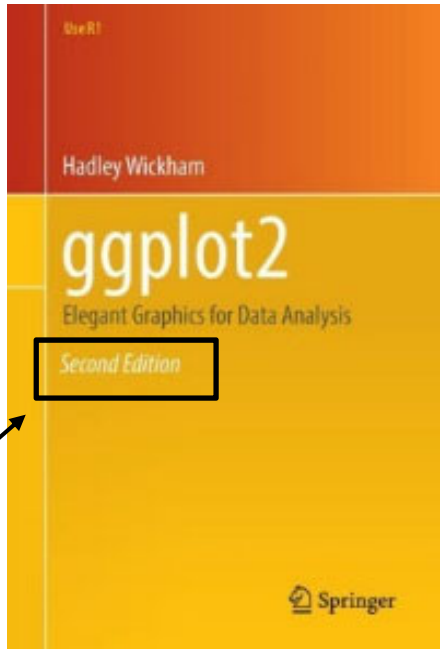
- From https://ggplot2.tidyverse.org

R Package: ggplot2

Used to produce statistical graphics, author = Hadley Wickham

"attempt to take the good things about base and lattice graphics and improve on them with a **strong, underlying model "**

described in ***ggplot2*** *Elegant Graphs for Data Analysis, Second Edition*, 2016

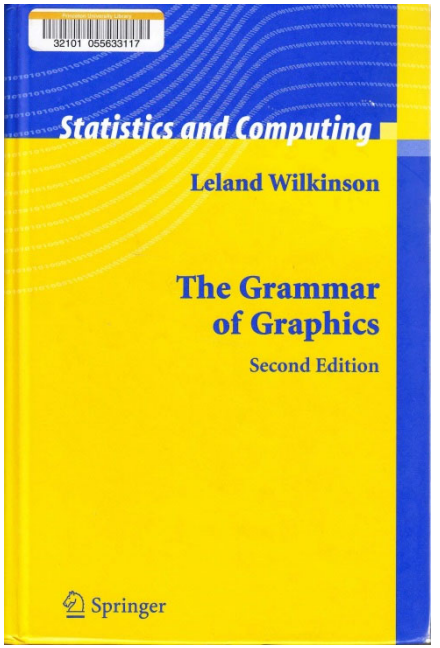


based on *The Grammar of Graphics* by Leland Wilkinson, 2005

"... describes *the meaning* of what we do when we construct statistical graphics ... More than a taxonomy ... Computational system based on the underlying mathematics of representing statistical functions of data."

- does not limit developer to a set of pre-specified graphics

adds some concepts to grammar which allow it to work well with R



qplot()

ggplot2 provides two ways to produce plot objects:

qplot() # quick plot – not covered in this workshop

uses some concepts *of The Grammar of Graphics*, but doesn't provide full capability

and

designed to be very similar to plot() and simple to use

may make it easy to produce basic graphs

but

may delay understanding philosophy of ggplot2

ggplot() # grammar of graphics plot – focus of this workshop

provides fuller implementation of *The Grammar of Graphics*

may have steeper learning curve but allows much more flexibility when building graphs

Grammar Defines Components of Graphics

data: in ggplot2, data must be stored as an R data frame

coordinate system: describes 2-D space that data is projected onto
- for example, Cartesian coordinates, polar coordinates, map projections, ...

geoms: describe type of geometric objects that represent data
- for example, points, lines, polygons, ...

aesthetics: describe visual characteristics that represent data
- for example, position, size, color, shape, transparency, fill

scales: for each aesthetic, describe how visual characteristic is converted to display values
- for example, log scales, color scales, size scales, shape scales, ...

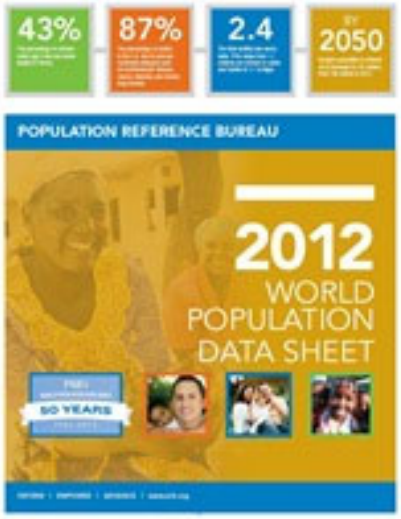
stats : describe statistical transformations that typically summarize data
- for example, counts, means, medians, regression lines, ...

facets: describe how data is split into subsets and displayed as multiple small graphs

Workshop Data Frame

extract from 2012 World Population Data Sheet produced by Population Reference Bureau

includes data for 158 countries where mid-2012 population >= 1 million



variables:

`country` country name
`pop2012` population mid-2012 (millions)
`imr` infant mortality rate*
`tfr` total fertility rate*
`le` life expectancy at birth
`leM` male life expectancy at birth
`leF` female life expectancy at birth
`area` (Africa, Americas, Asia & Oceania, Europe)
`region` (Northern Africa, Western Africa, Eastern Africa, Middle Africa, North America, Central America, Caribbean, South America, Western Asia, South Central Asia, Southeast Asia, East Asia, Oceania, Northern Europe, Western Europe, Eastern Europe, Southern Europe)

*definitions: infant mortality rate – annual number of deaths of infants under age 1 per 1,000 live births
total fertility rate – average number of children a woman would have assuming that current age-specific birth rates remain constant throughout her childbearing years

Create a Plot Object

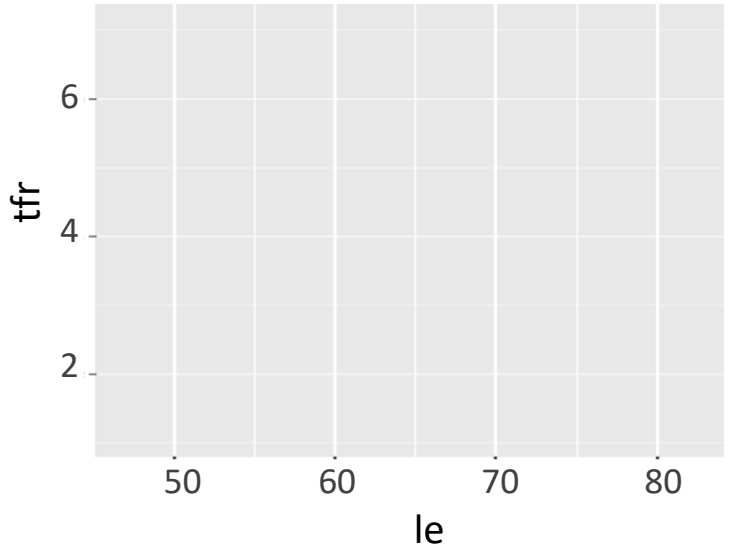
creates a **plot object** that can be assigned to a variable

can specify data frame and aesthetic mappings (visual characteristics that represent data)

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
p
```

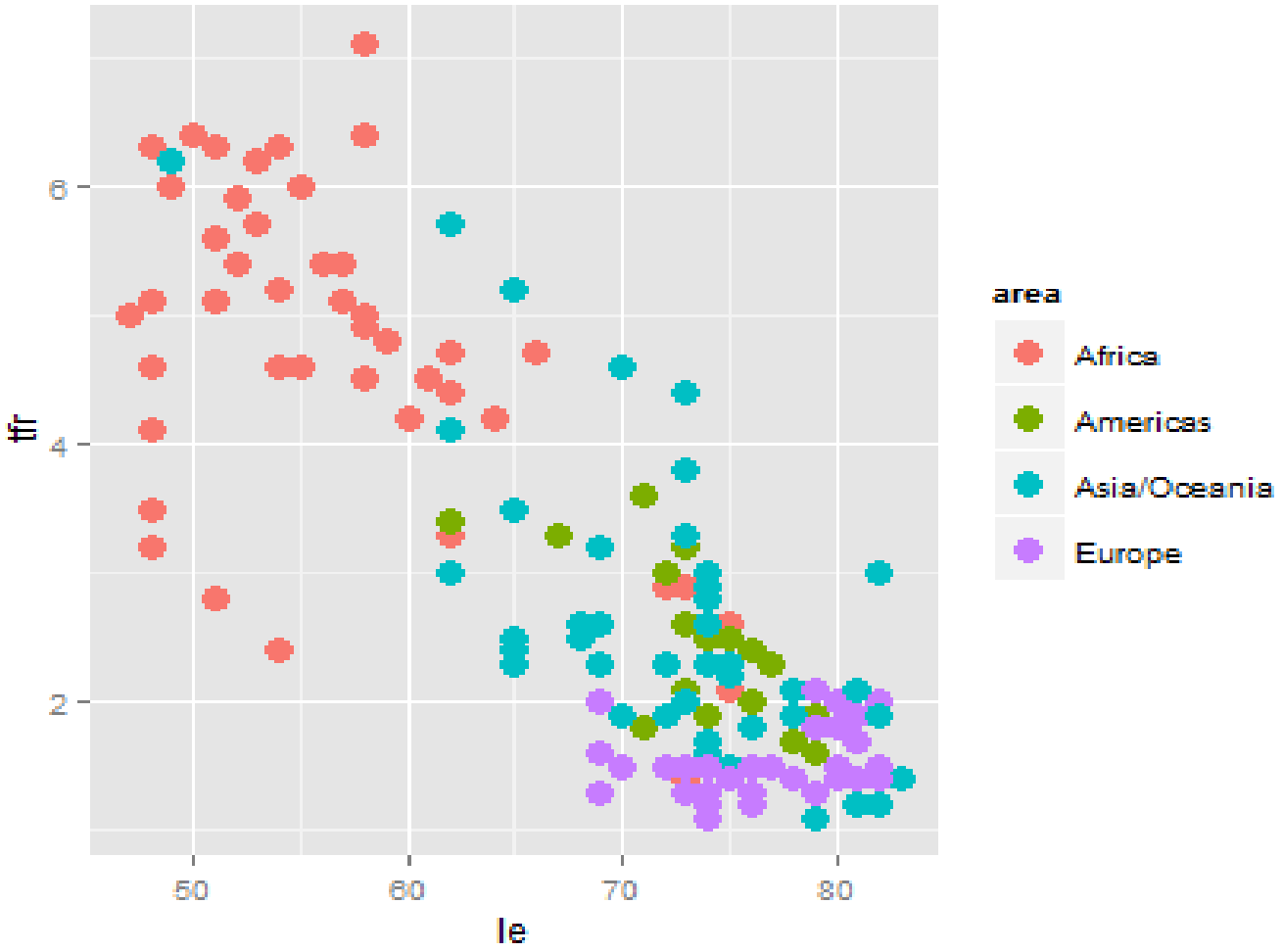
country	pop2012	tfr	le	area
Algeria	37.4	2.9	73	Africa
Egypt	82.3	2.9	72	Africa
Libya	6.5	2.6	75	Africa
Morocco	32.6	2.3	72	Africa
South Sudan	9.4	5.4	52	Africa
Sudan	33.5	4.2	60	Africa
Tunisia	10.8	2.1	75	Africa
Benin	9.4	5.4	56	Africa
Burkina Faso	17.5	6.0	55	Africa
Cote d'Ivoire	20.6	4.6	55	Africa
Gambia	1.8	4.9	58	Africa
Ghana	25.5	4.2	64	Africa
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

x-axis position indicates le value
y-axis position indicates tfr value



Adding a Layer

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
p + geom point(size=4)
```



Layer

purpose:

display the data –

allows viewer to see patterns, overall structure, local structure, outliers, ...

display statistical summaries of the data –

allows viewer to see counts, means, medians, IQRs, model predictions, ...

data and *aesthetics* (mappings) may be inherited from ggplot() object or added, changed, or dropped within individual layers

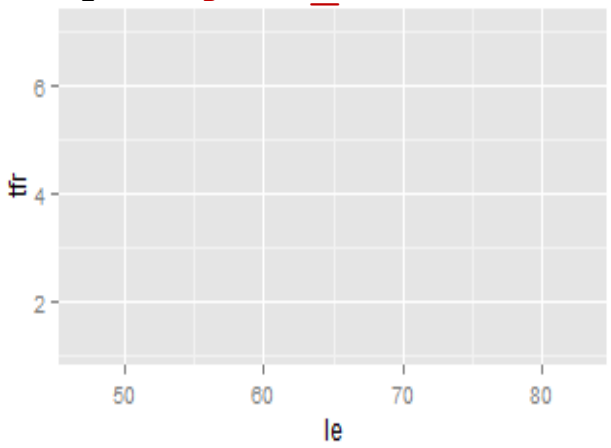
most layers contain a geom ... the fundamental building block of ggplot2
full specification: geom_xxx(mapping, data, stat, position, ...)

each geom_xxx() has a default stat (statistical transformation) associated with it, but the default statistical transformation may be changed using stat parameter

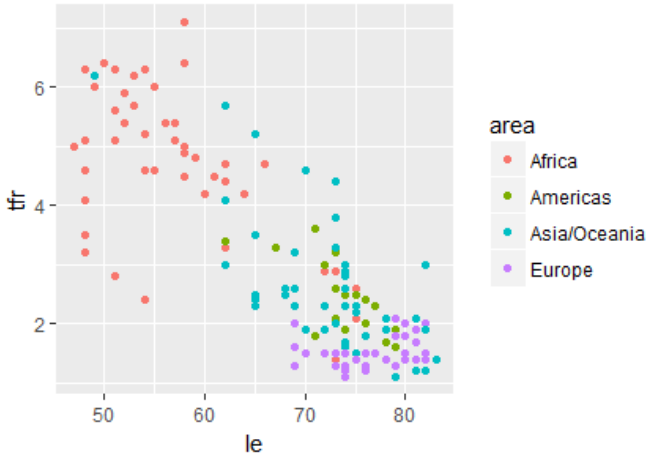
Adding a *geom* Layer

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
```

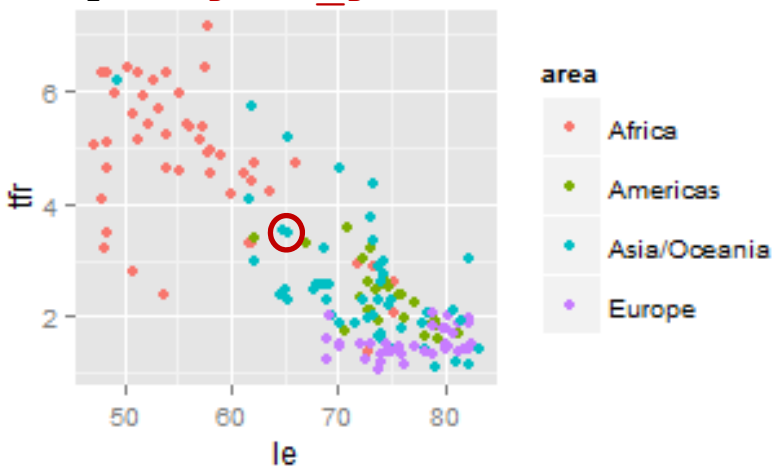
```
p + geom_blank()
```



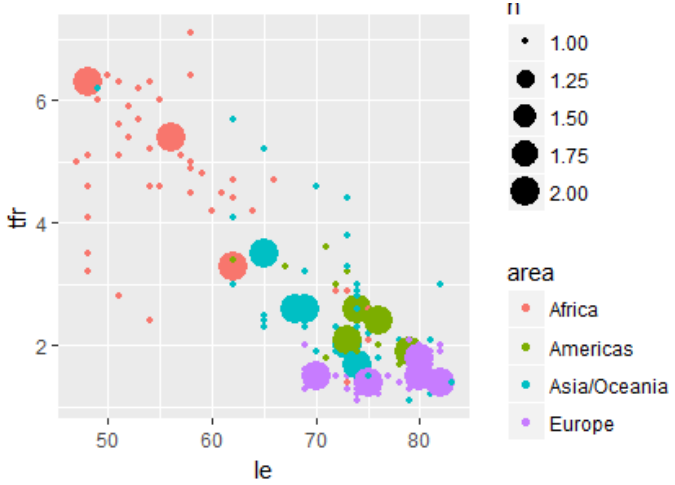

```
p + geom_point()
```




```
p + geom_jitter()
```



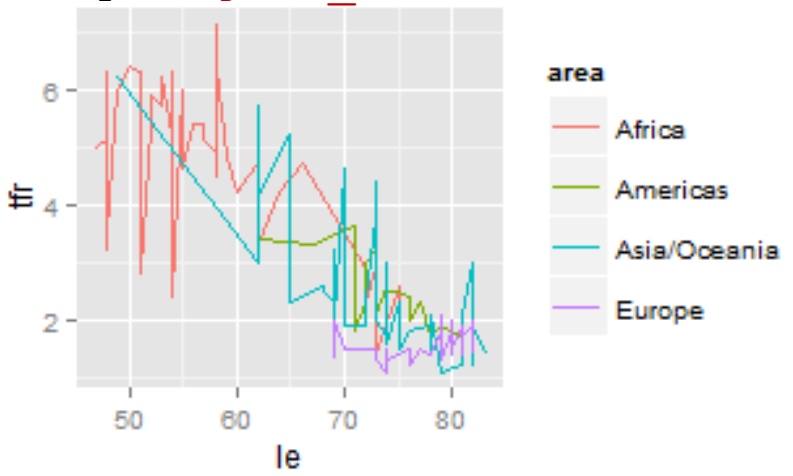

```
p + geom_count()
```



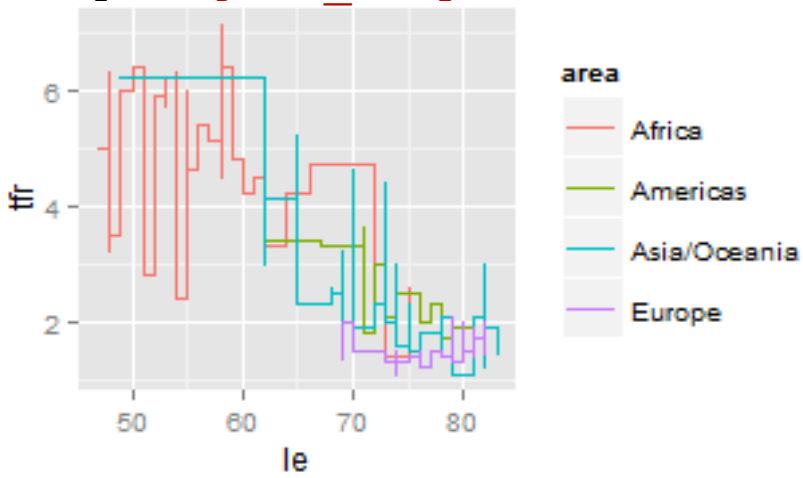
Adding a *geom* Layer: Connect Points

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
```

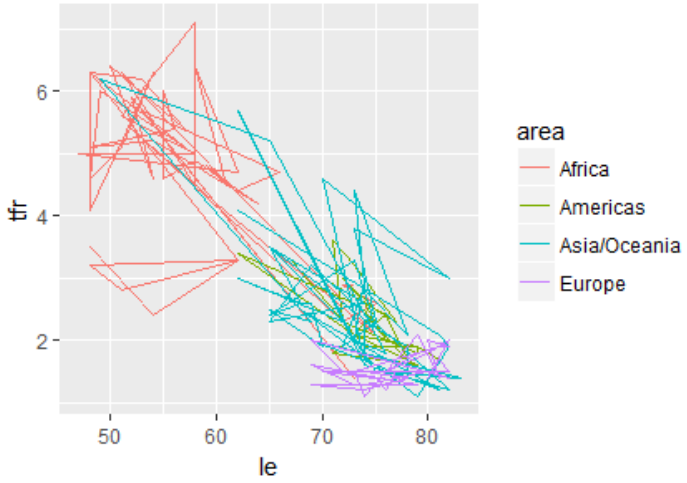
p + geom_line()



p + geom_step()

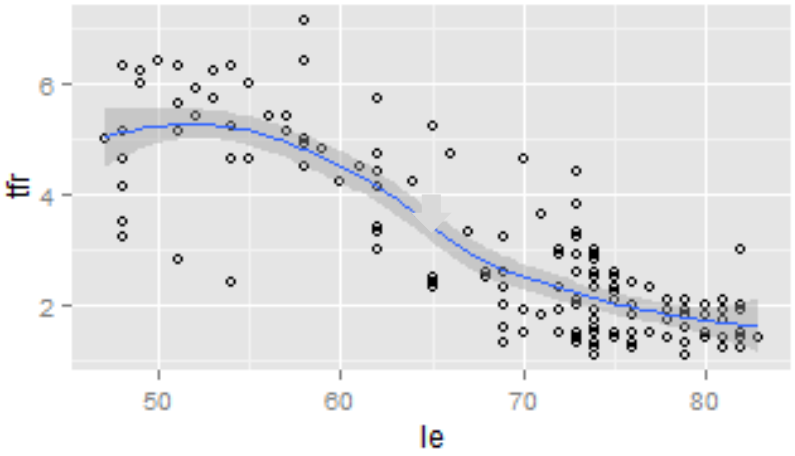


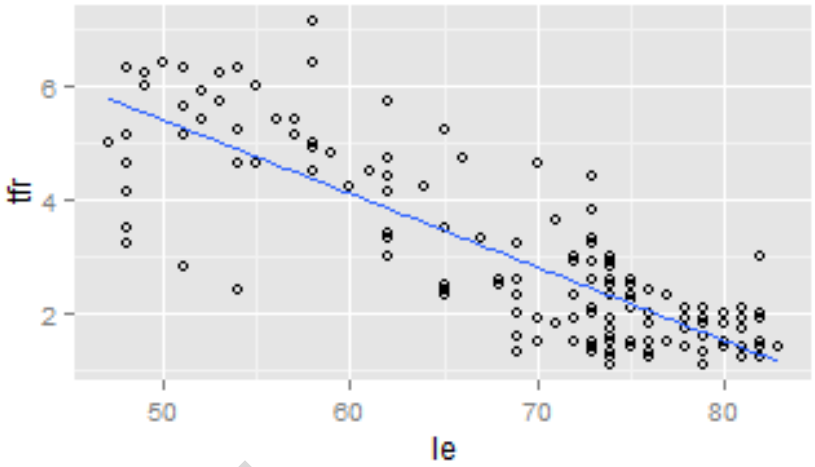
p + geom_path()



Displaying Data and Statistical Summary

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
p + geom point(shape=1) + geom_smooth()
```

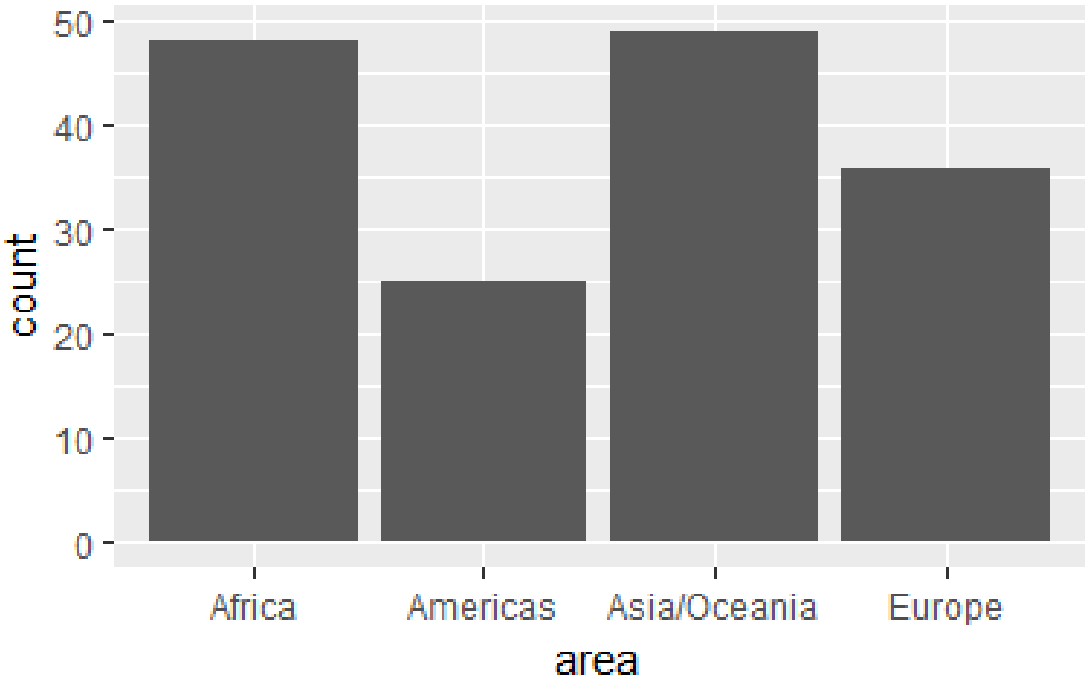





```
p + geom_point(shape=1) + geom_smooth(method="lm", se=FALSE)
```

Displaying Statistical Summary

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=area))
p + geom_bar()
```



Already Transformed Data

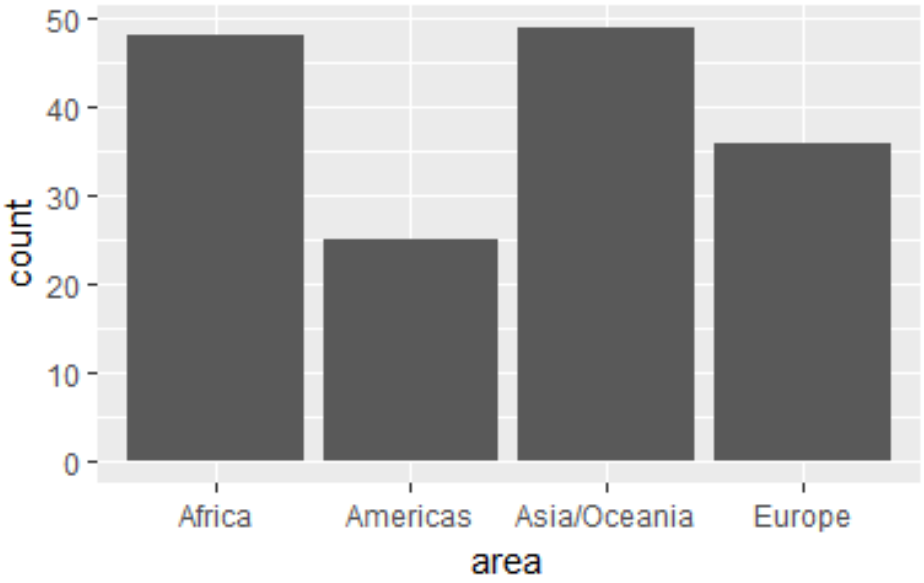
```
wb <- read.csv(file="WDS2012areabins.csv", head=TRUE, sep=",")
wb
```

	bin	area	count
1	1	Africa	48
2	2	Americas	25
3	3	Asia/Oceania	49
4	4	Europe	36

```
p <- ggplot(data=wb, aes(x=area, y=count))
p + geom_col()
#       OR
p + geom_bar(stat="identity")
```

geom_bar: height of bar proportional to number of observations in each group.
geom_col: leaves data as is.

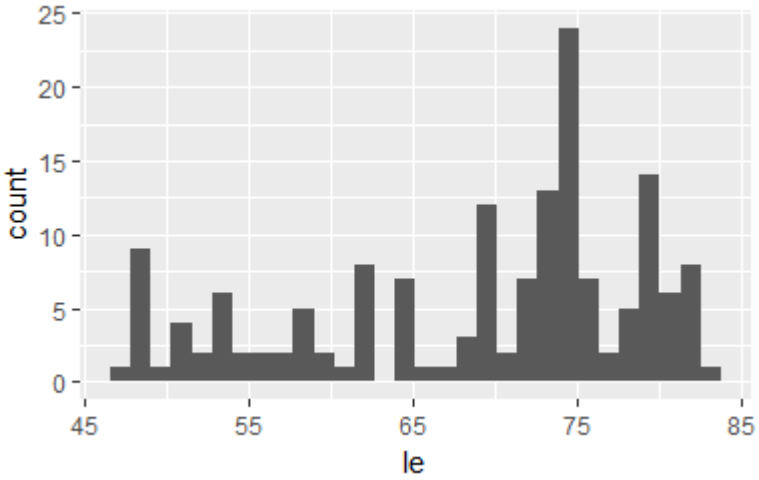
geom_bar uses count stat by default.
geom_col uses identity stat.



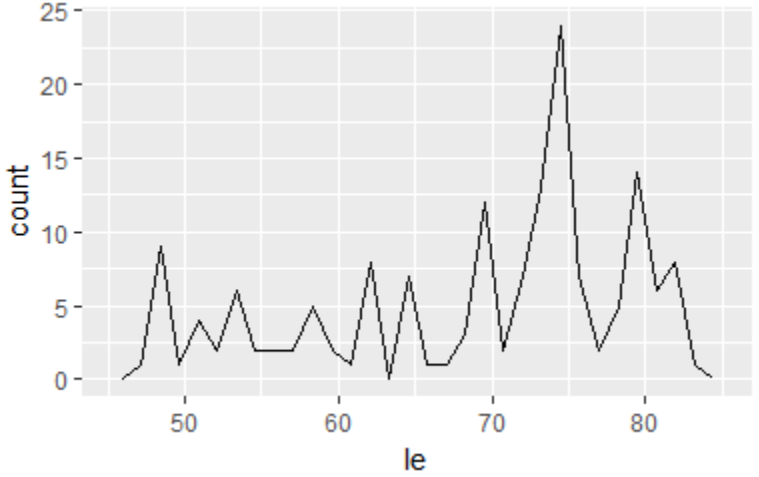
Displaying Distributions

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le))
```

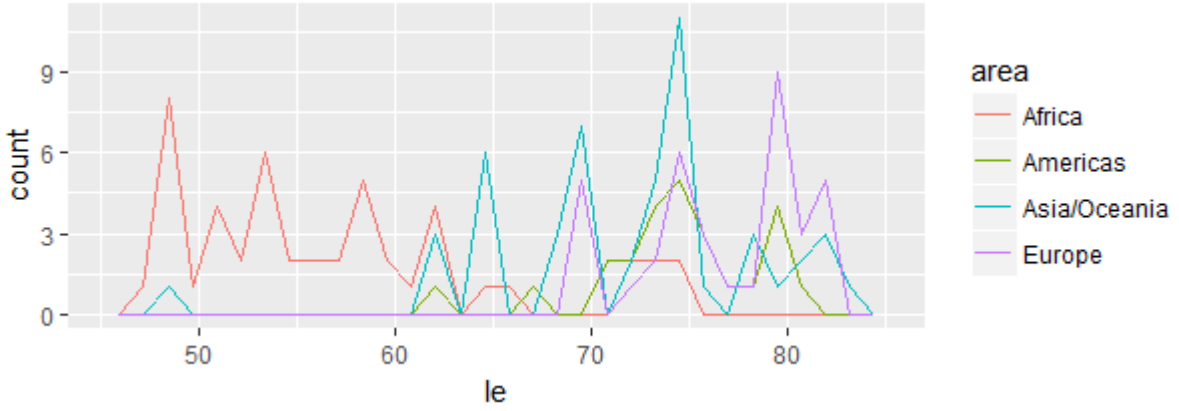
```
p + geom_histogram()
```




```
p + geom_freqpoly()
```



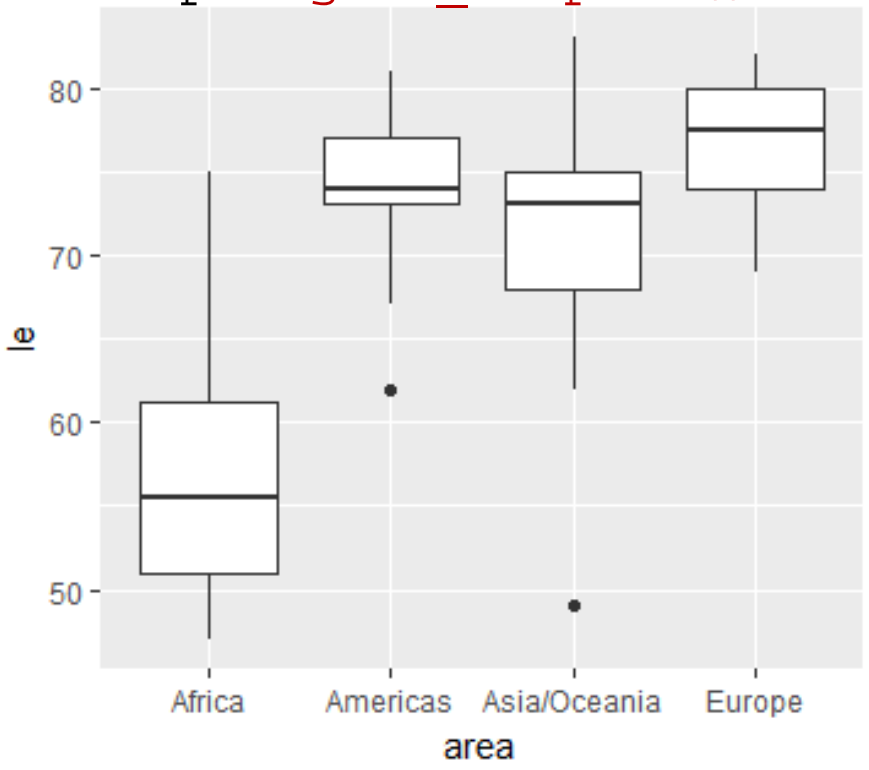

```
p + geom_freqpoly(aes(color=area))
```



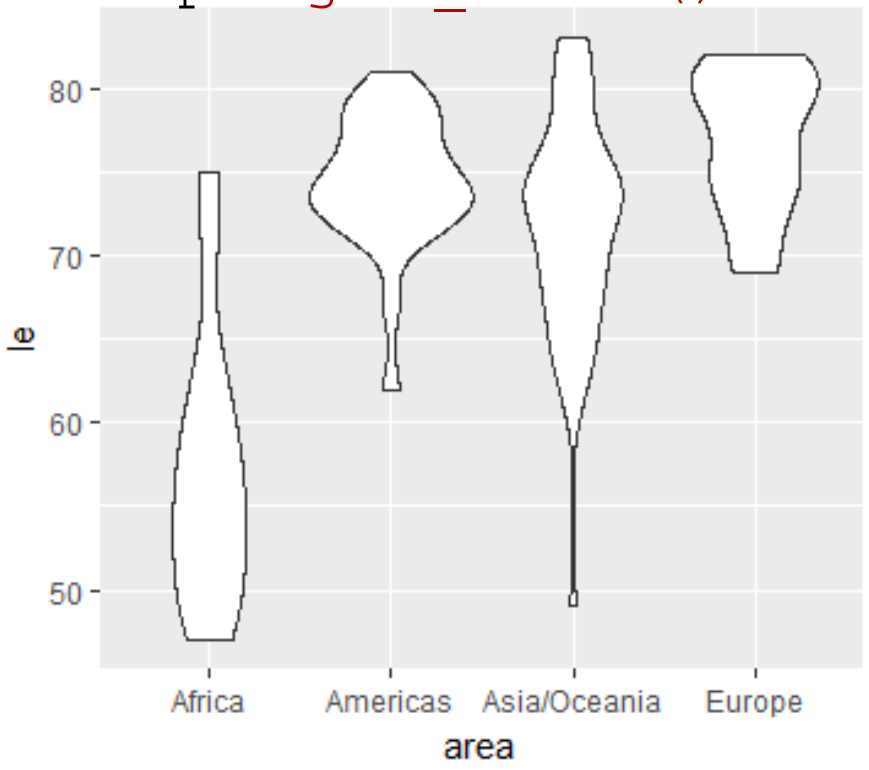
Displaying Statistical Summaries

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=area, y=le))
```

p + geom_boxplot()



p + geom_violin()



geoms

graphical primitives

geom_blank

geom_curve

geom_path

geom_polygon

geom_rect

geom_ribbon

geom_abline
geom_hline
geom_vline

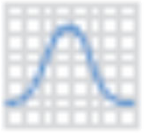
geom_segment
geom_spoke

one variable, discrete

geom_bar

one variable, continuous

geom_area



geom_density

geom_dotplot

geom_freqpoly

geom_histogram

geom_qq

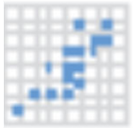
For more detail, see Rstudio's Cheat Sheet, Data Visualization with ggplot2, at https://www.rstudio.com/resources/cheatsheets/

geoms

two variables, both continuous

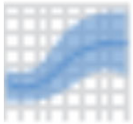
geom_label


geom_jitter


geom_point


geom_quantile

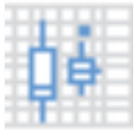

geom_rug


geom_smooth

geom_text

two variables, discrete x, continuous y


geom_col


geom_boxplot


geom_dotplot


geom_violin

two variables, discrete x, discrete y


geom_count

For more detail, see Rstudio's Cheat Sheet, Data Visualization with ggplot2, at https://www.rstudio.com/resources/cheatsheets/

geoms

two variables, visualizing error


geom_crossbar


geom_errorbar
geom_errorbarh


geom_linerange


geom_pointrange

two variables,
continuous bivariate distribution


geom_bin2d

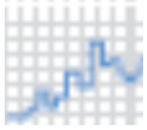

geom_density2d


geom_hex

two variables, continuous function


geom_area


geom_line


geom_step

two variables, maps


geom_map

Full specification of each geom at:
http://ggplot2.tidyverse.org/reference/#section-layer-geoms

For more detail, see Rstudio's Cheat Sheet, Data Visualization with ggplot2, at https://www.rstudio.com/resources/cheatsheets/

Aesthetics

describe visual characteristics that represent data

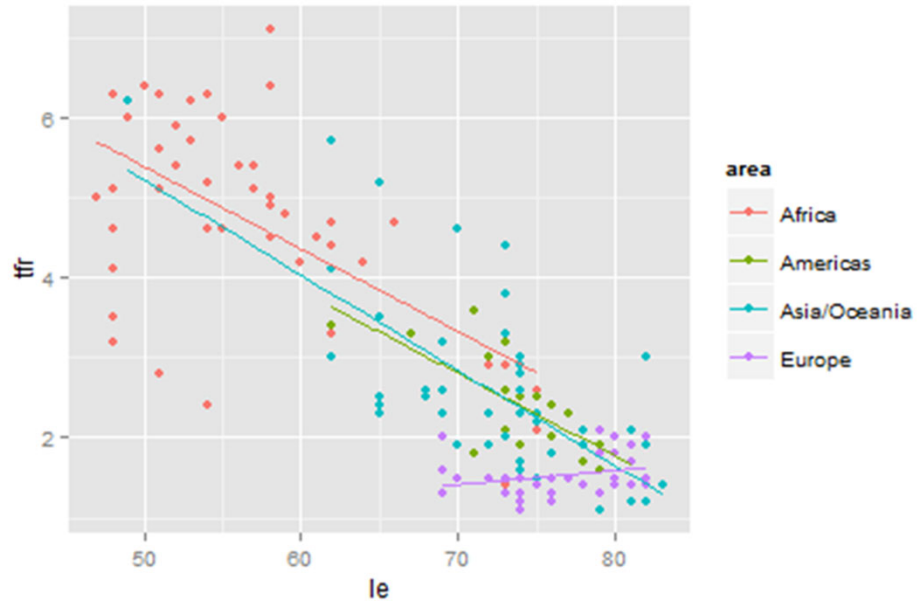
- for example, x position, y position, size, color (outside), fill (inside), point shape, line type, transparency

each layer inherits default aesthetics from plot object

- within each layer, aesthetics may added, overwritten, or removed

most layers have some required aesthetics and some optional aesthetics

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
p + geom_point() + geom_smooth(method="lm", se=FALSE)
```

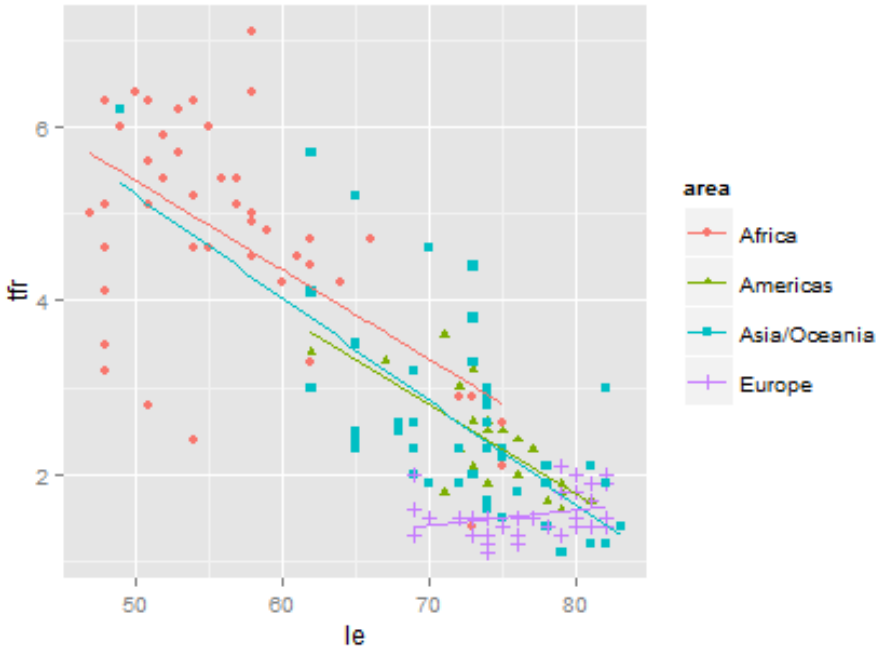


Add or Remove Aesthetic Mapping

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
```

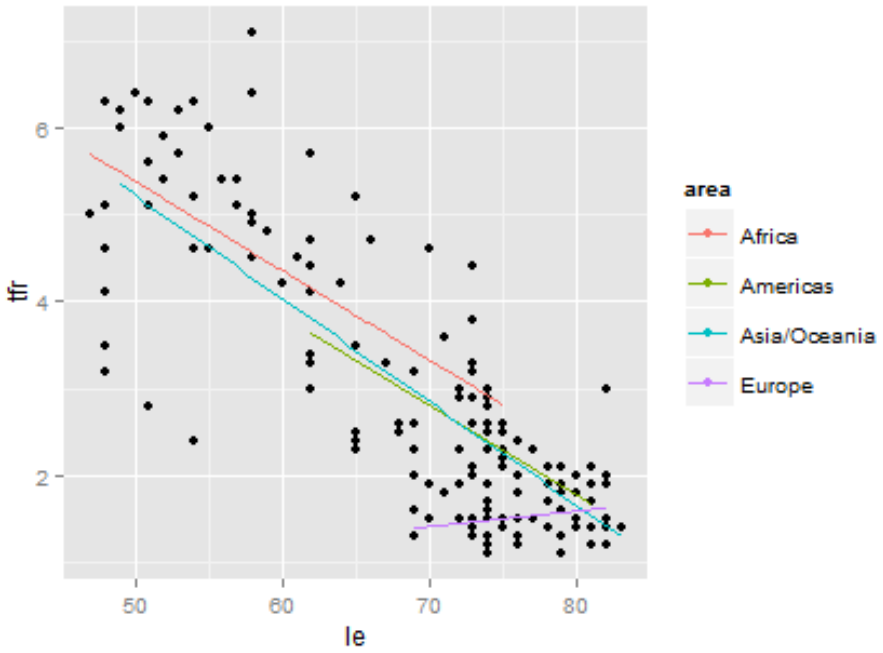
add aesthetic mapping

```
p + geom_point(aes(shape=area)) +
geom_smooth(method="lm",se=FALSE)
```



remove aesthetic mapping

```
p + geom_point(aes(color=NULL)) +
geom_smooth(method="lm", se=FALSE)
```



Aesthetic Mapping vs. Parameter Setting

aesthetic mapping
- data value determines visual characteristic
- use *aes()*

setting
- constant value determines visual characteristic
- use layer parameter

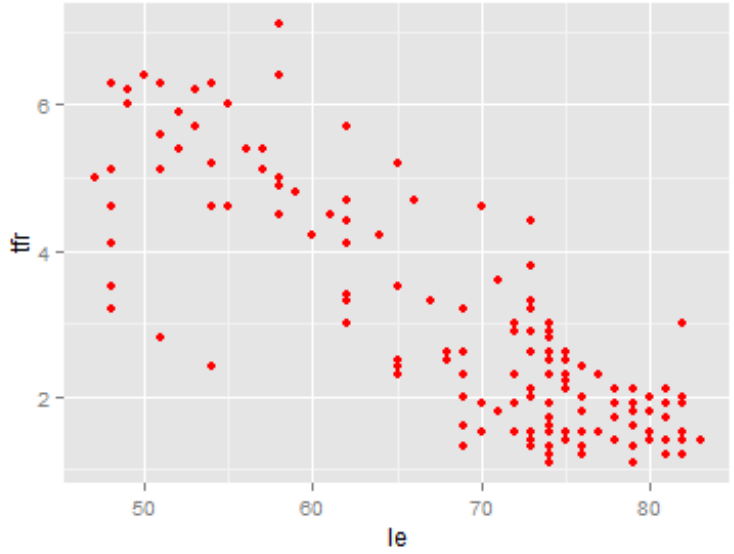
```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
```

aesthetic mapping

```
p + geom_point(aes(color=area))
```

setting

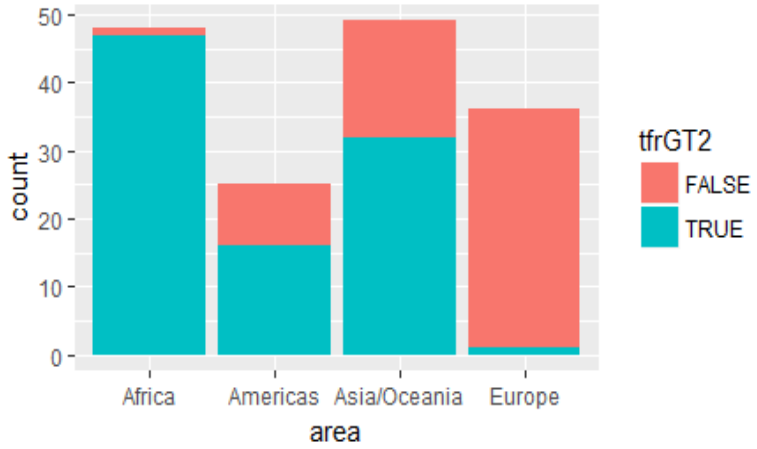
```
p + geom_point(color="red")
```



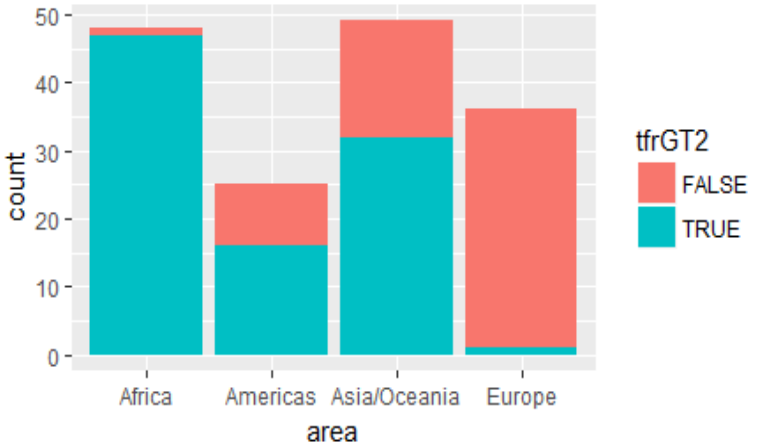
Position

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w, aes(x=area, fill=tfrGT2))
```

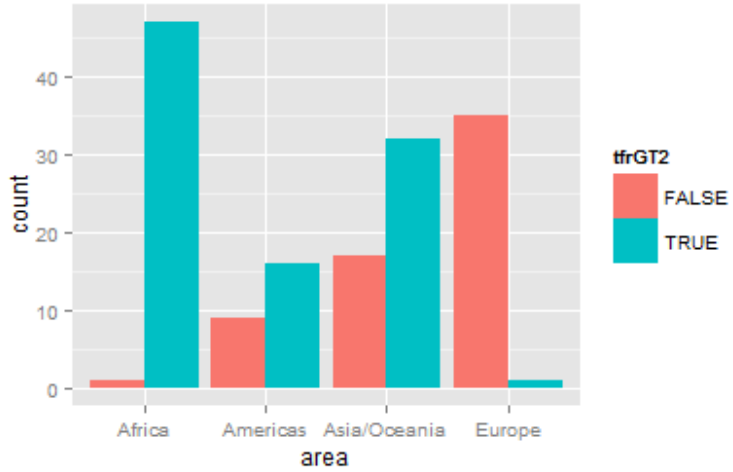
```
p + geom_bar()
```



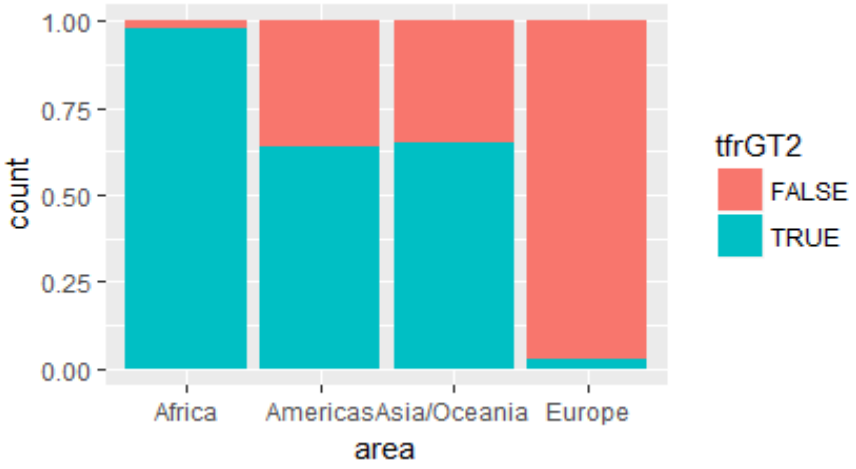

```
p + geom_bar(position="stack")
```




```
p + geom_bar(position="dodge")
```



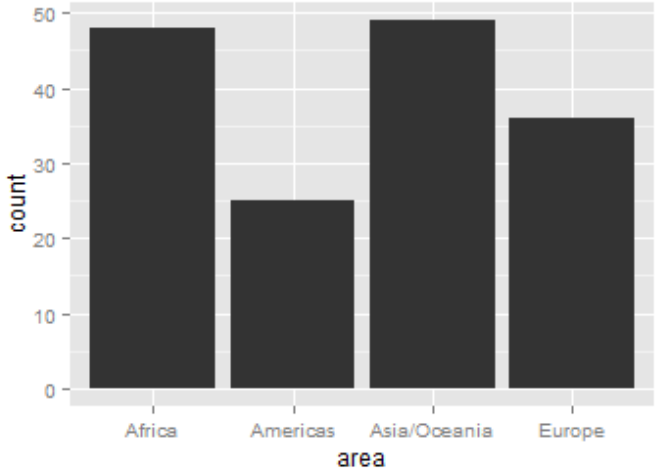

```
p + geom_bar(position="fill")
```



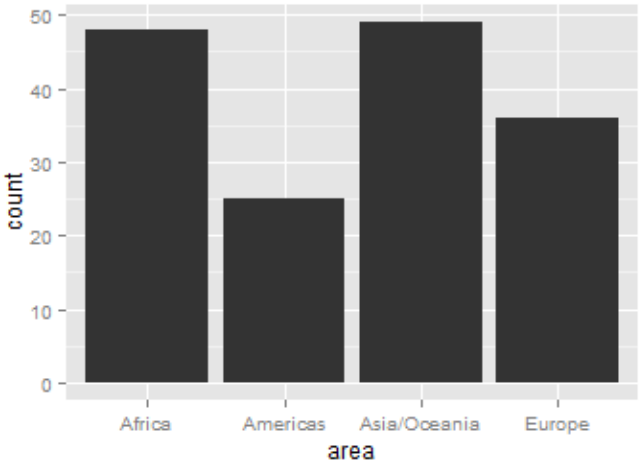
Bar Width

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=area))
```

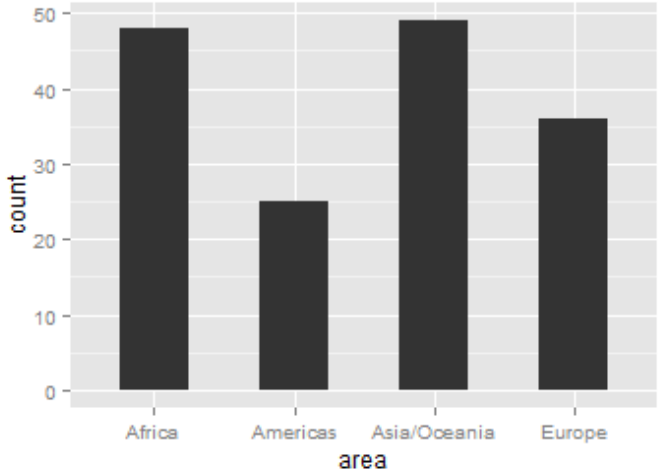
```
p + geom_bar()
```



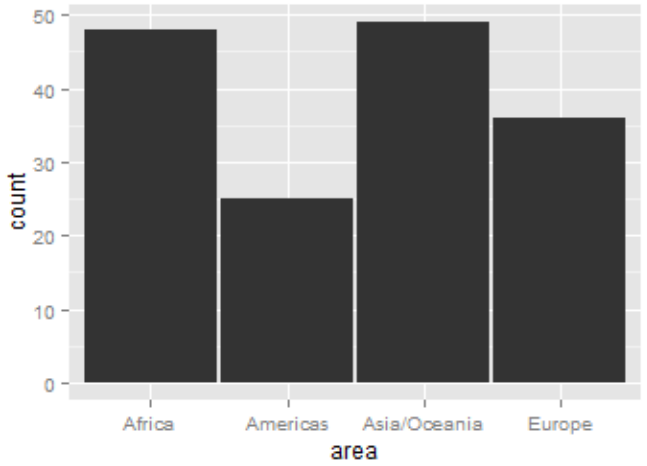

```
p + geom_bar(width=.9) # default
```




```
p + geom_bar(width=.5)
```

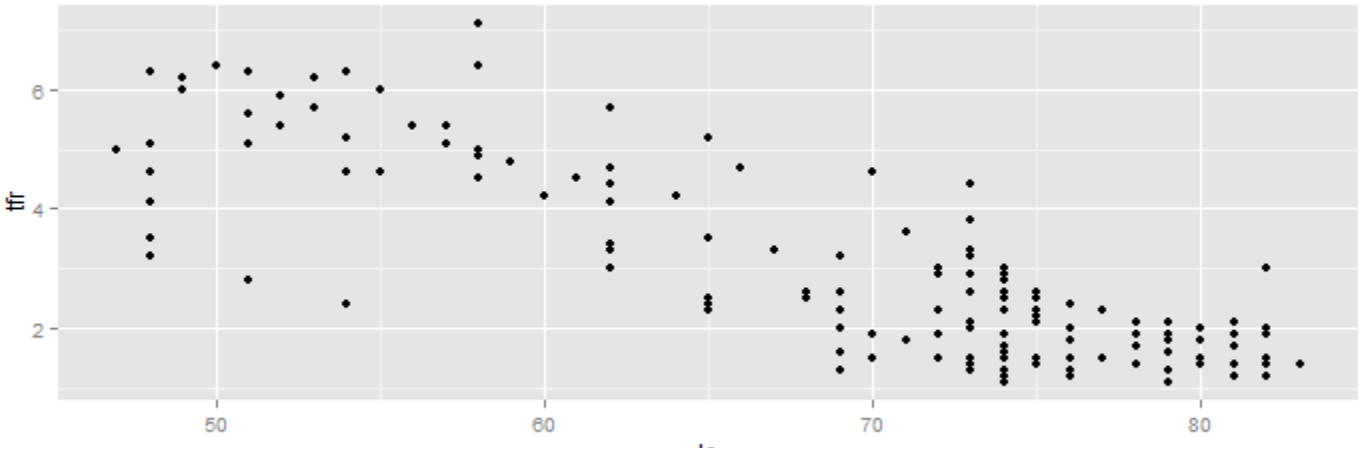



```
p + geom_bar(width=.97)
```

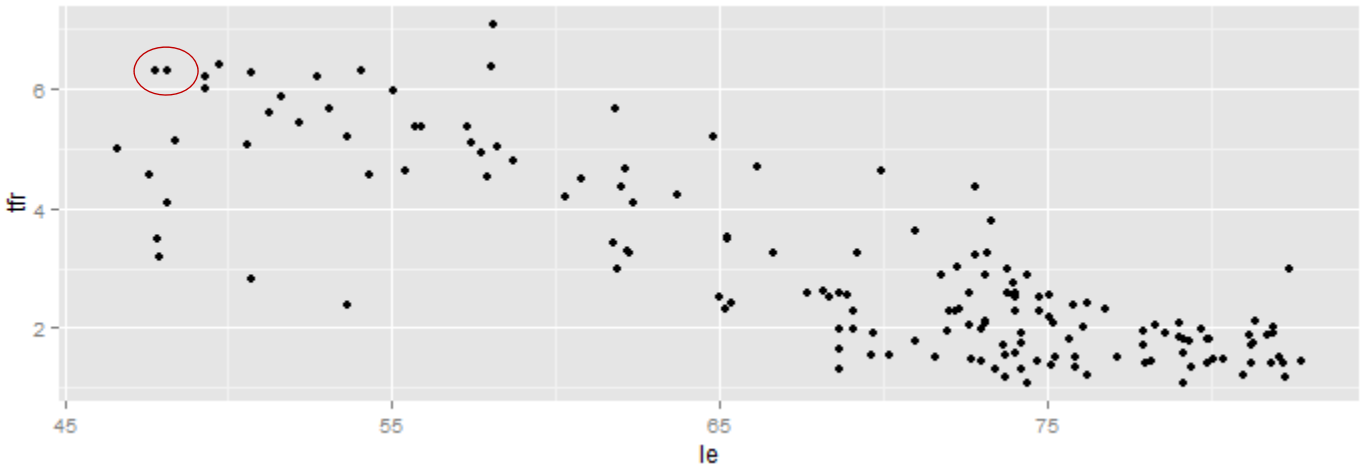


Position

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
```




```
p + geom_point()
```




```
p + geom_point
(position="jitter")
```

equivalent to

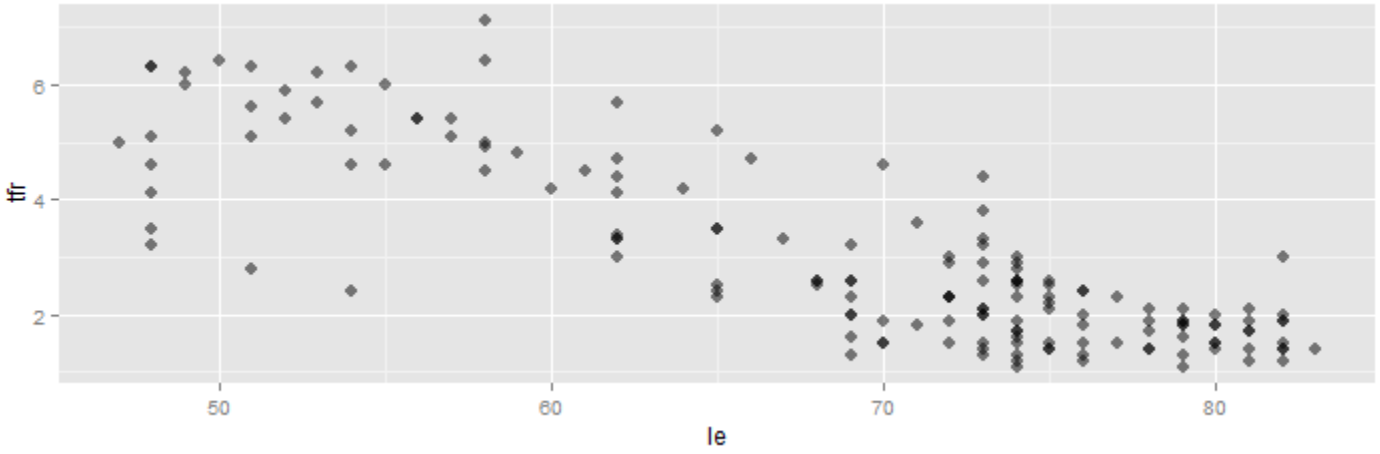
```
p + geom_jitter()
```

For reproducible jittering, set a seed ... new "seed" argument for **`position_jitter()`**, as of 3.1.0:

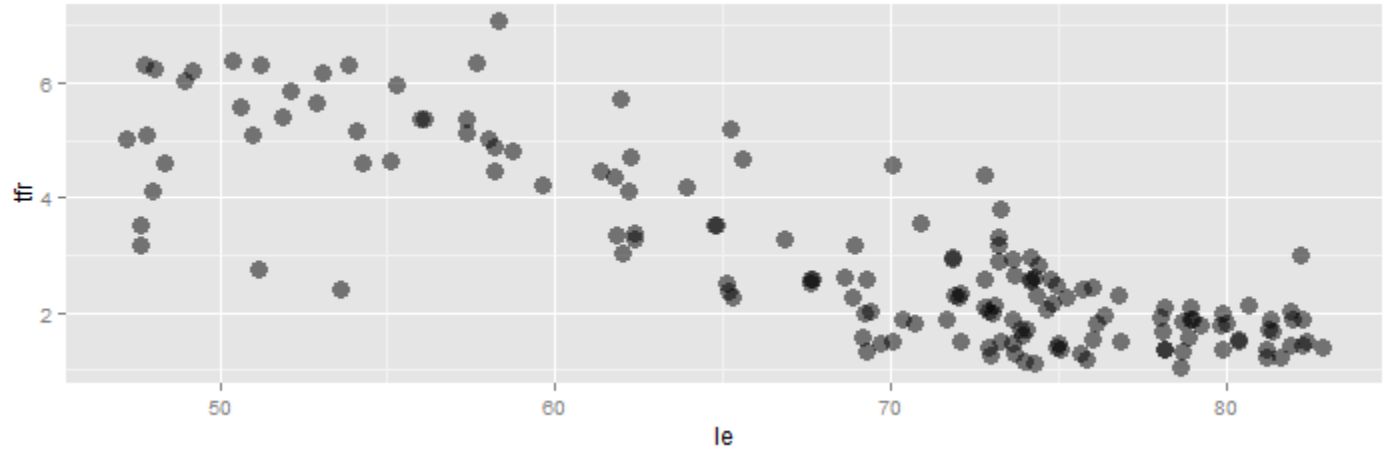
```
p + geom_point(position=position_jitter(seed=1))
```

Transparency

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
```




```
p + geom_point
(size=3,
alpha=1/2)
```



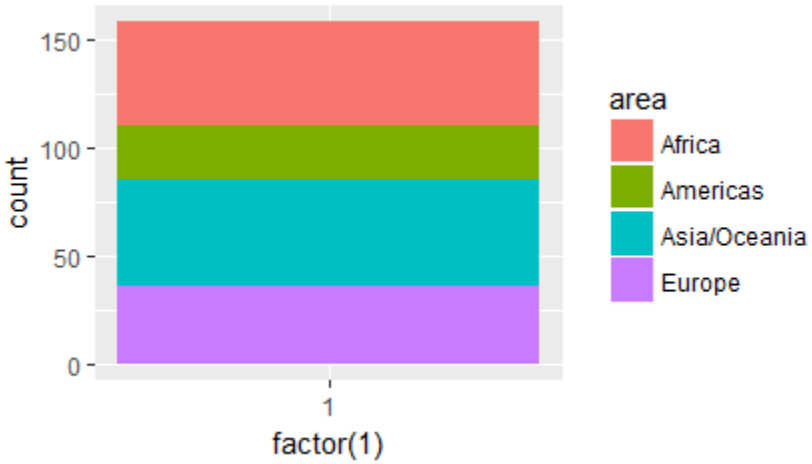

```
p + geom_jitter
(size=4,
alpha=1/2)
```

techniques for overplotting: adjusting symbol size, shape, jitter, and transparency

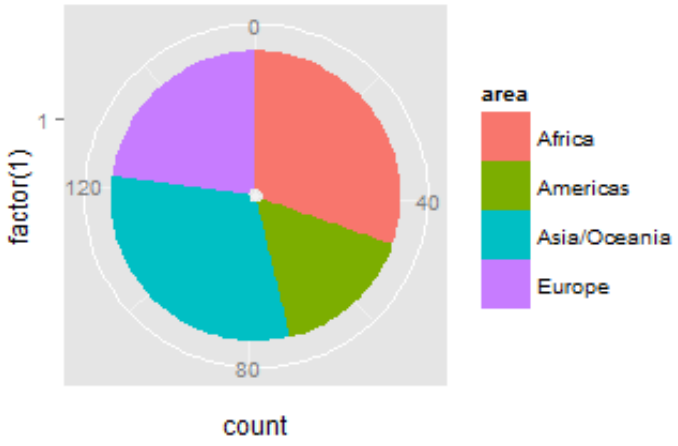
Coordinate System

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(w, aes(x=factor(1), fill=area))
```

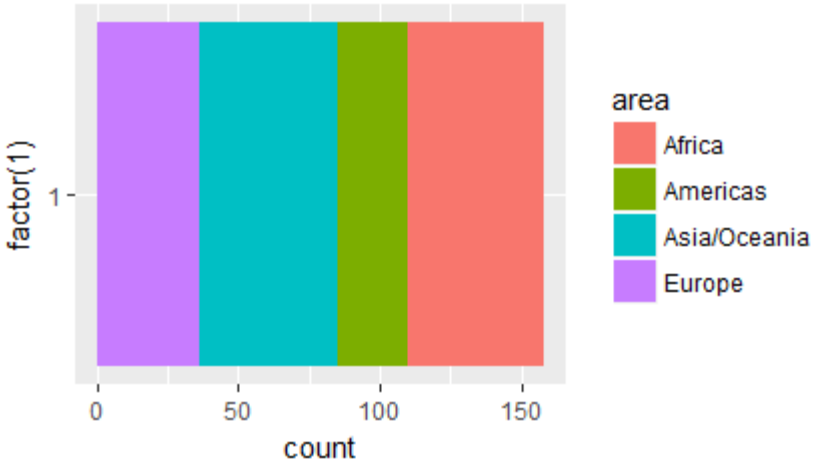
```
p + geom_bar()
```



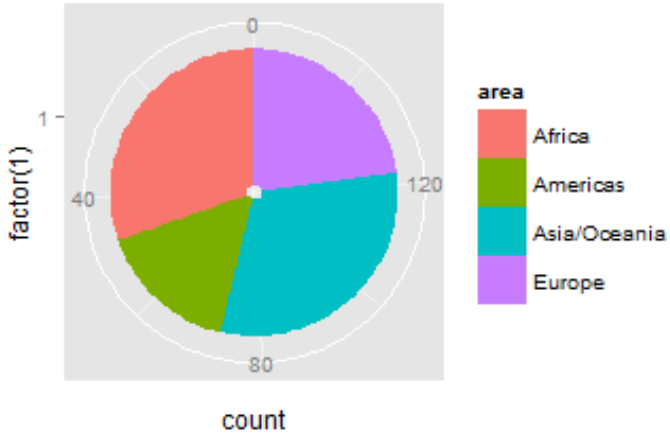

```
p + geom_bar() + coord_polar(theta="y")
```




```
p + geom_bar() + coord_flip()
```



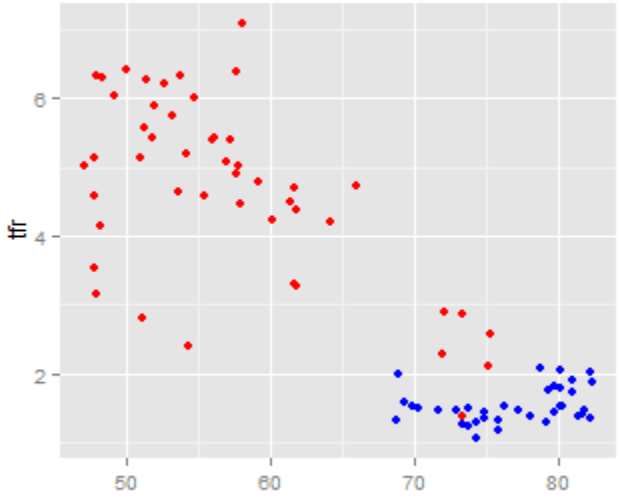

```
p + geom_bar() + coord_polar(theta="y",
                             direction=-1)
```



Data Frame

each plot layer may contain data from a different data frame

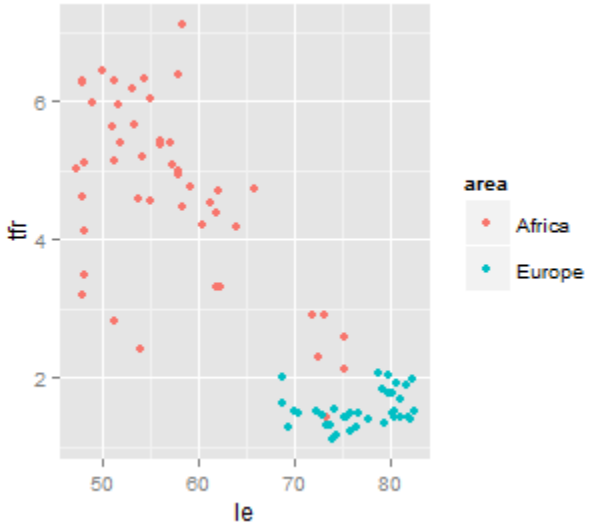
```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
africa <- subset(w,area=="Africa")
europe <- subset(w,area=="Europe")

p <- ggplot(data=europe, aes(x=le, y=tfr))
p + geom_jitter(color="blue") +
    geom_jitter(data=africa, color="red")
```




```
africa_europe <- rbind(africa, europe)
p <- ggplot(data=africa_europe, aes(x=le, y=tfr,
            color=area))
p + geom_jitter()
```

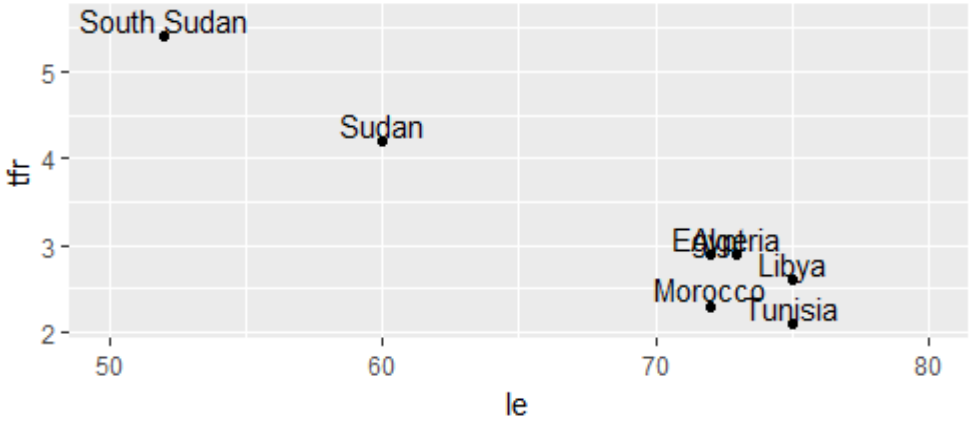
OR

```
p <- ggplot(data=rbind(africa,europe), aes(le, y=tfr,
            color=area))
p + geom_jitter()
```

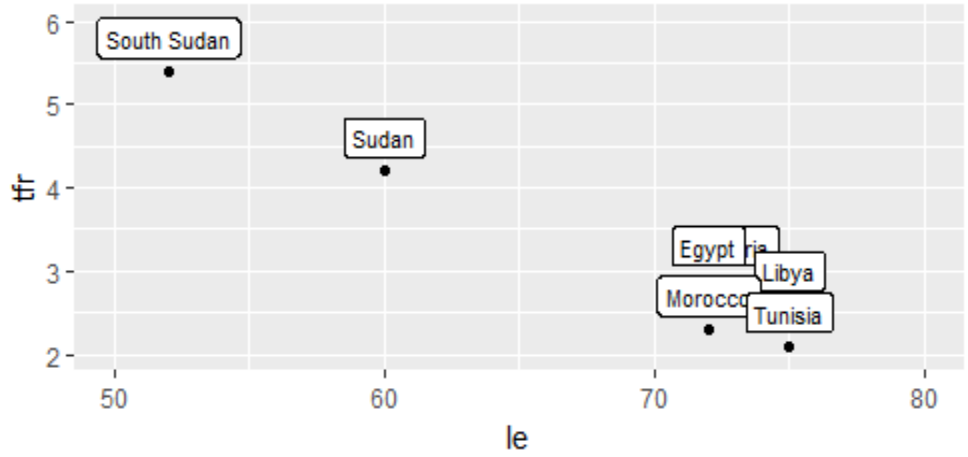


Labels

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
wna <- subset(w, region=="Northern Africa")
p <- ggplot(data=wna, aes(x=le, y=tfr))
```



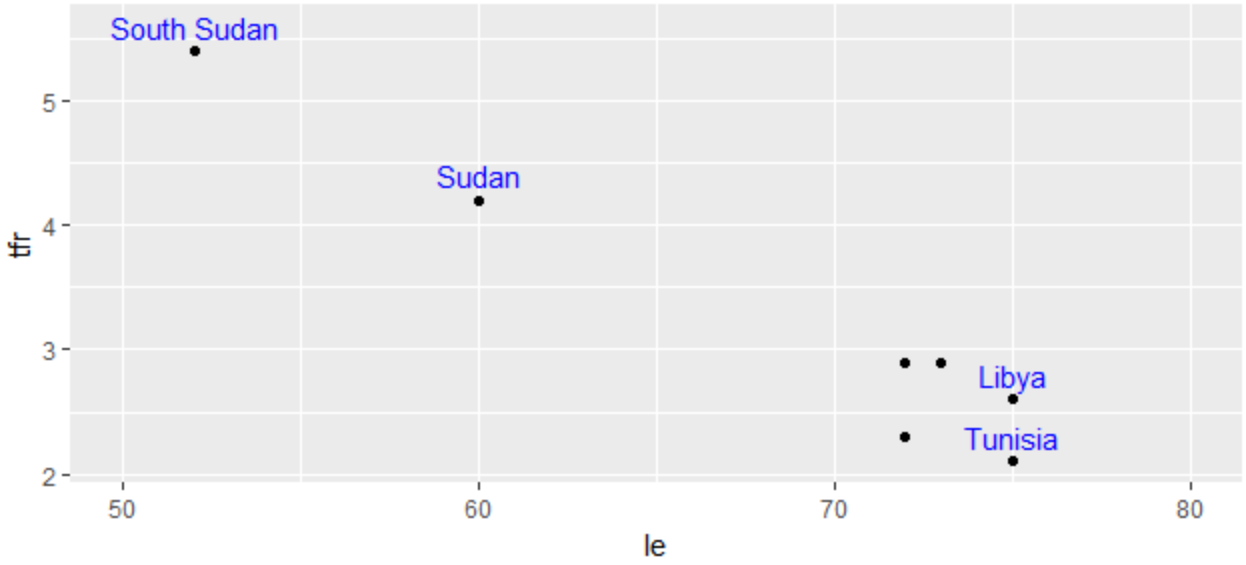

```
p + geom_point() +
geom_text(aes(label=country),
nudge_y=.2, size=4) +
xlim(50,80)
```




```
p + geom_point() +
geom_label(aes(label=country),
nudge_y=.3, size=3) +
xlim(50,80) + ylim(2,6)
```

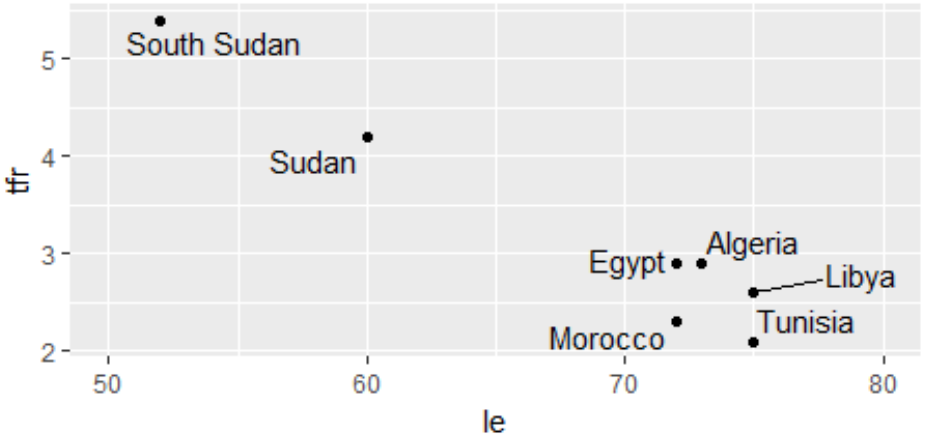
Labels

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
labelset <-c("South Sudan", "Sudan", "Libya", "Tunisia")

p <- ggplot(data=subset(w,region=="Northern Africa"),
     aes(x=le, y=tfr))
p +
geom_point() +
geom_text(data=subset(w, country %in% labelset),
             aes(label=country), nudge_y = .2, color="blue") +
xlim(50,80)
```

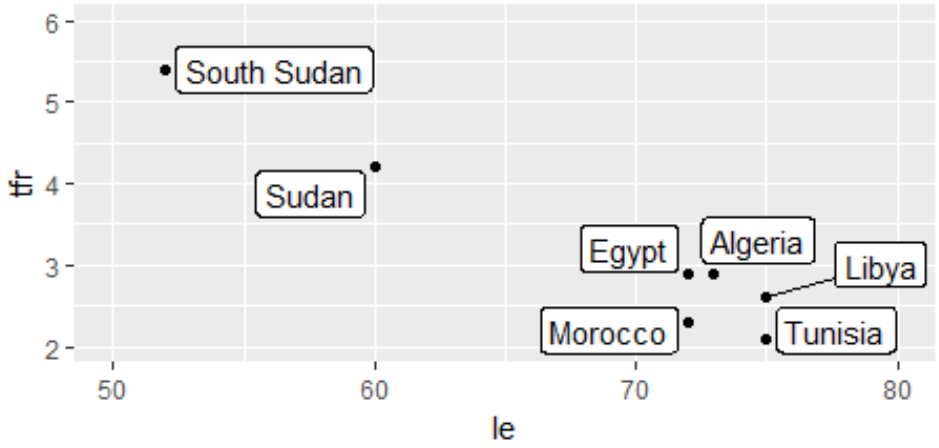


Non-Overlapping Labels

```
install.packages("ggrepel")
library("ggrepel")
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
wna <- subset(w, region=="Northern Africa")
p <- ggplot(data=wna, aes(x=le, y=tfr))
```



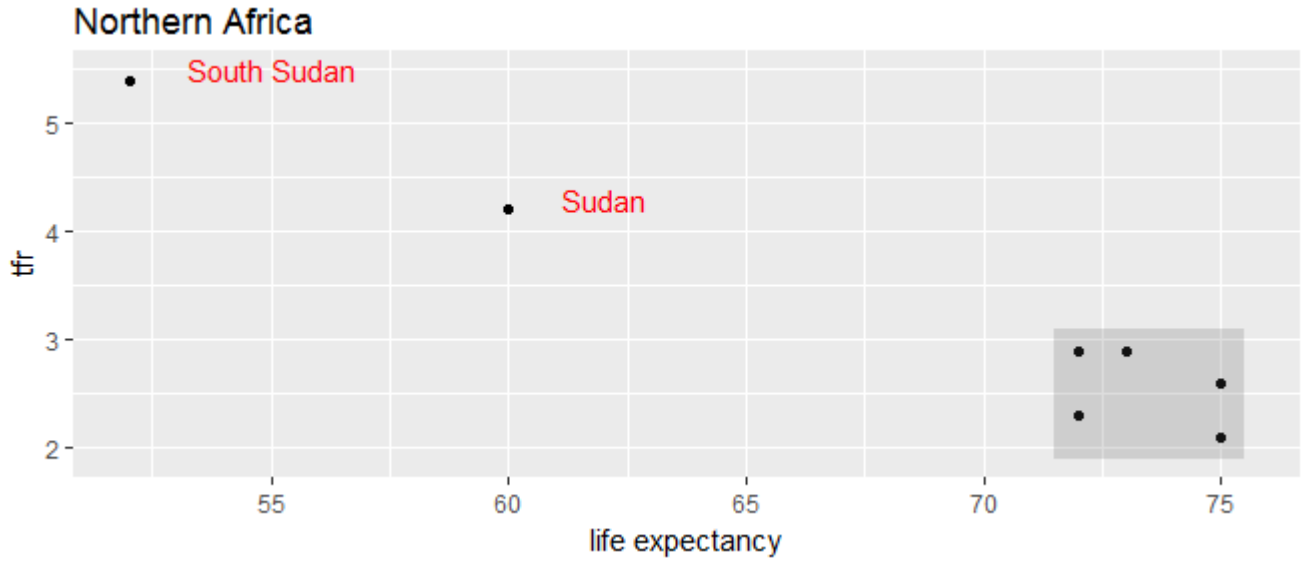

```
p + geom_point() +
  geom_text_repel(aes(
  label=country), size=4) +
  xlim(50,80)
```




```
p + geom_point() +
  geom_label_repel(aes(
  label=country), size=4) +
  xlim(50,80) + ylim(2,6)
```

Annotations

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
wna <- subset(w, region=="Northern Africa")
p <- ggplot(data=wna, aes(x=le, y=tfr))

p + geom_point() +
    annotate("text", x=55, y=5.5,  label="South Sudan", color="red") +
    annotate("text", x=62, y=4.3,  label="Sudan", color="red") +
    annotate("rect", xmin = 71.5, xmax = 75.5, ymin = 1.9, ymax = 3.1,
             alpha = .2) +
    ggtitle("Northern Africa") + xlab("life expectancy")
```



Scale

controls the mapping from data to aesthetic

"takes data and turns it into something that can be perceived visually"

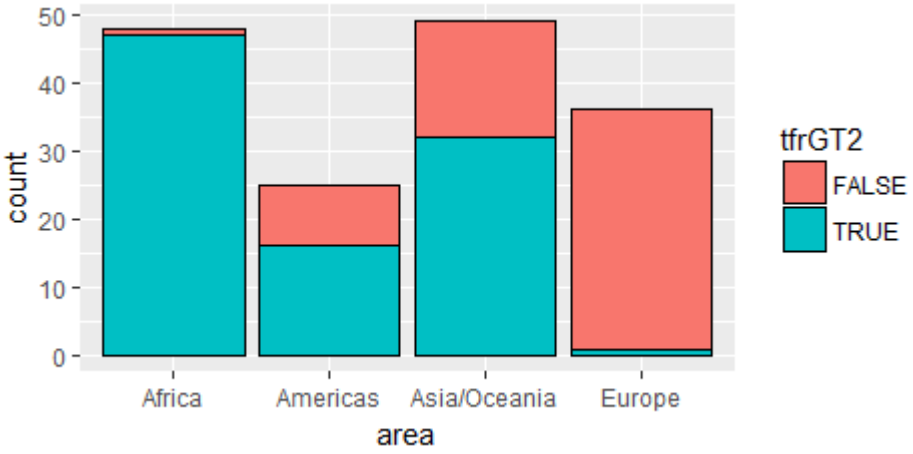
color and fill, shape, size, position

acts as a function from the data space to a place in the aesthetic space

provides axes or legends ("guides") to allow viewer to perform inverse mapping from aesthetic space back to data space

required for every aesthetic ... so ggplot2 always provides a default scale

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w, aes(x=area, fill=tfrGT2))
```




```
p + geom_bar(color="black")
```

equivalent to

```
p + geom_bar(color="black") +
    scale_fill_discrete()
```

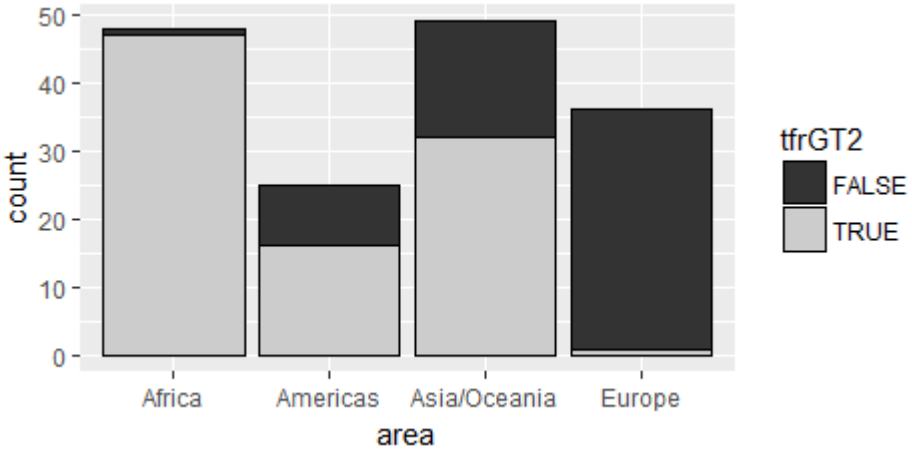
equivalent to

```
p + geom_bar(color="black") +
    scale_fill_hue()
```

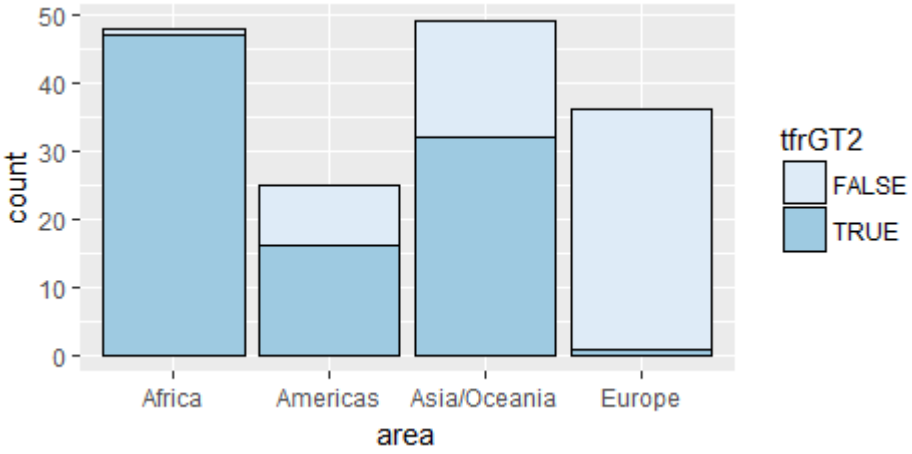
colors equally spaced around color wheel

Fill Scales

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w, aes(x=area, fill=tfrGT2))
```



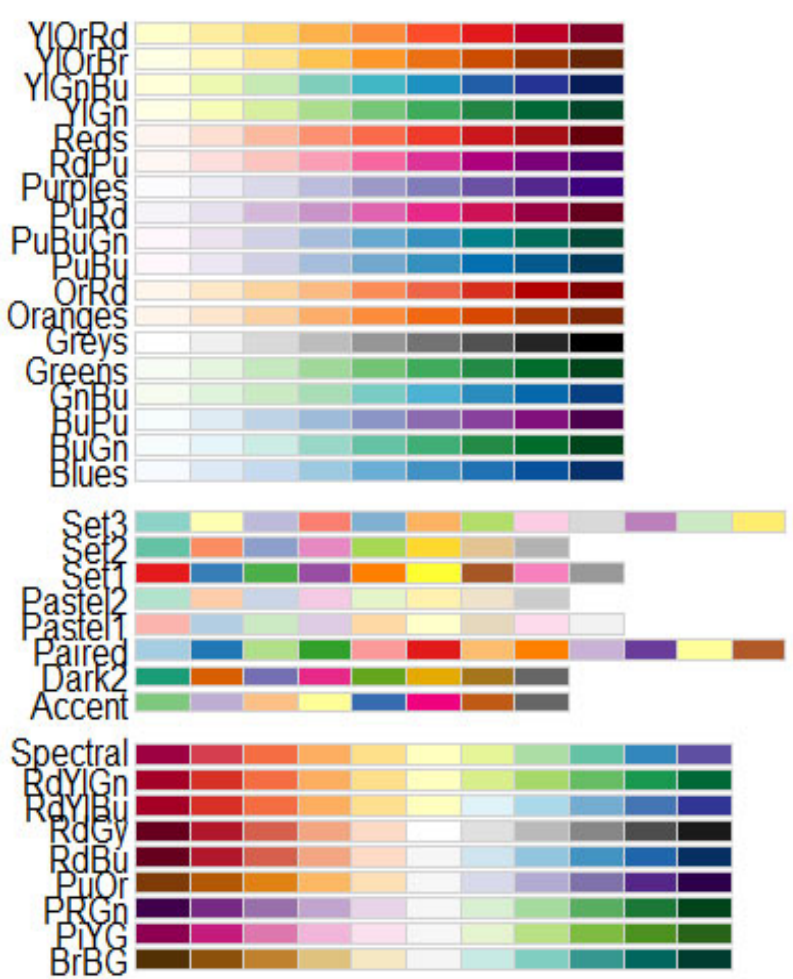

```
p + geom_bar(color="black") +
    scale_fill_grey()
```



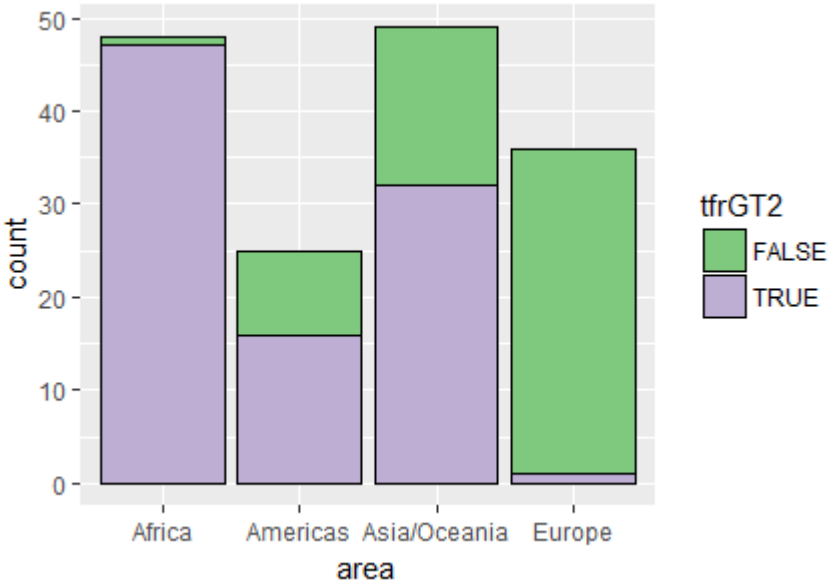

```
p + geom_bar(color="black") +
    scale_fill_brewer()
```

Fill Scales

```
library(RColorBrewer)
display.brewer.all()
```

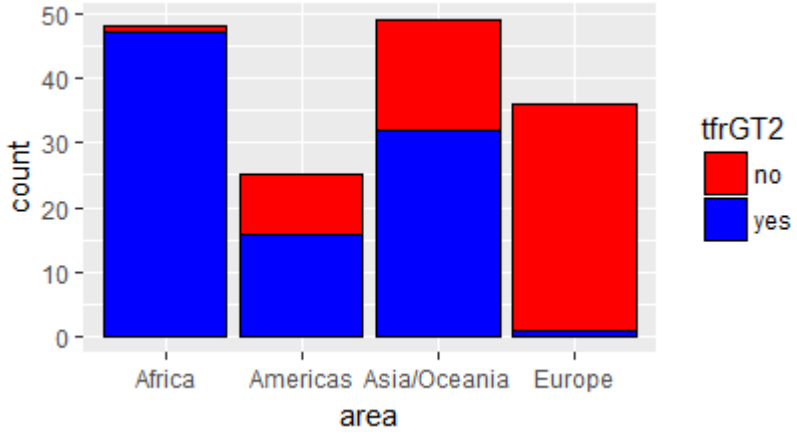



```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2

p <- ggplot(data=w,
     aes(x=area, fill=tfrGT2))

p + geom_bar(color="black") +
scale_fill_brewer(palette="Accent")
```



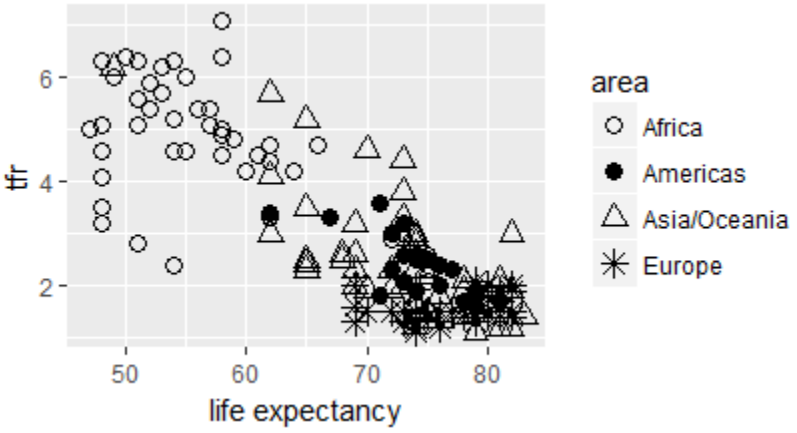
Manual Scales

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w, aes(x=area, fill=tfrGT2))
```




```
p + geom_bar(color="black") +
scale_fill_manual(values=c("red","blue"),
                  labels=c("no", "yes"))
```

typical scale arguments: values
labels
breaks
limits
name

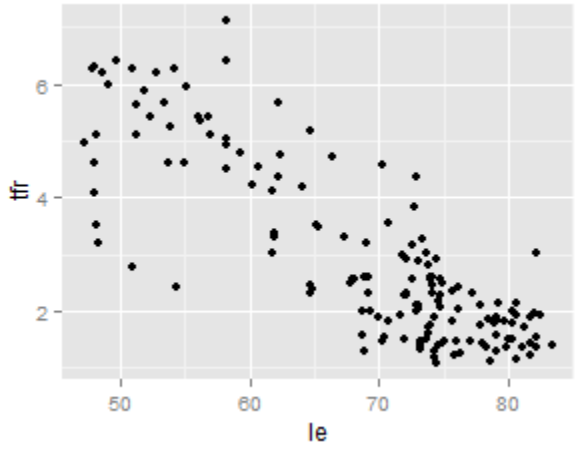



```
p + geom_point(aes(x=le, y=tfr,
    shape=area, fill=NULL), size = 3) +
    xlab("life expectancy") +
    scale_shape_manual(values=c(1,16,2,8))
```

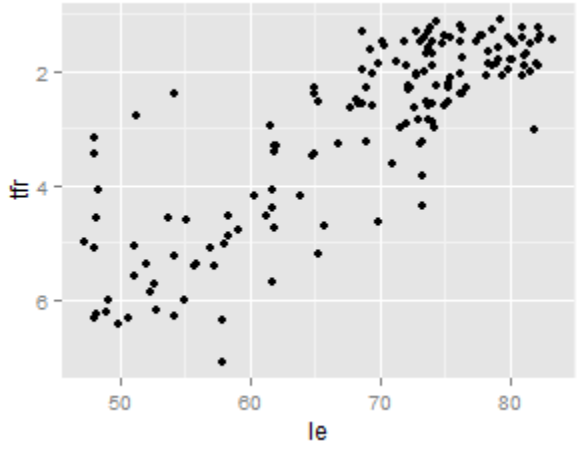
Position Scales

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
```

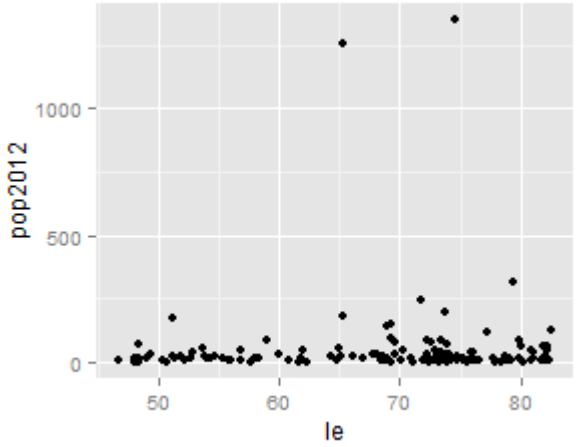
```
p <- ggplot(data=w, aes(x=le, y=tfr))
p + geom_jitter()
```



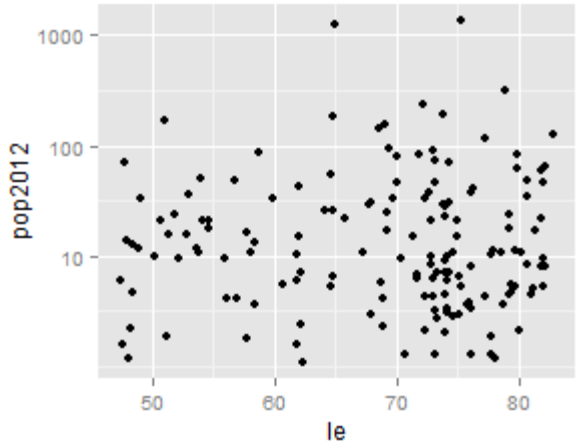

```
p + geom_jitter() +
    scale_y_reverse()
```




```
p <- ggplot(data=w,
            aes(x=le, y=pop2012))
p + geom_jitter()
```




```
p + geom_jitter() +
scale_y_log10(breaks=c(10, 100,
1000), labels=c(10,100,1000))
```

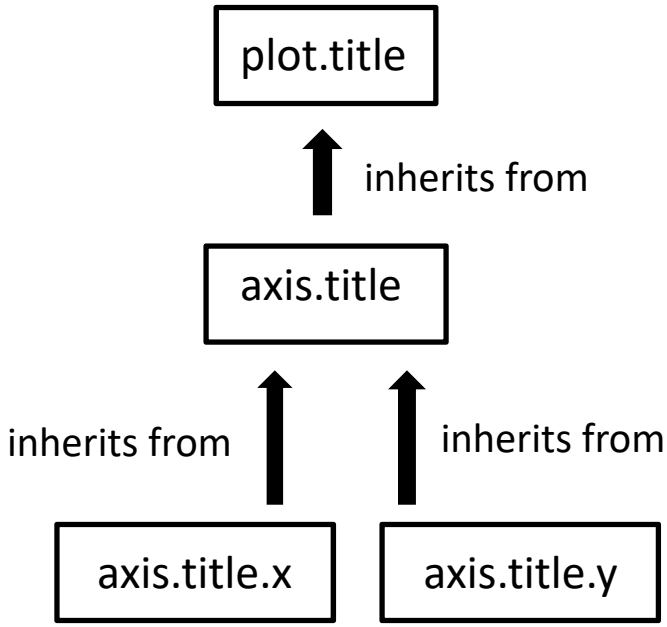


Theme

controls appearance of **non-data elements**
... does not affect how data is displayed by `geom_xxx()` function

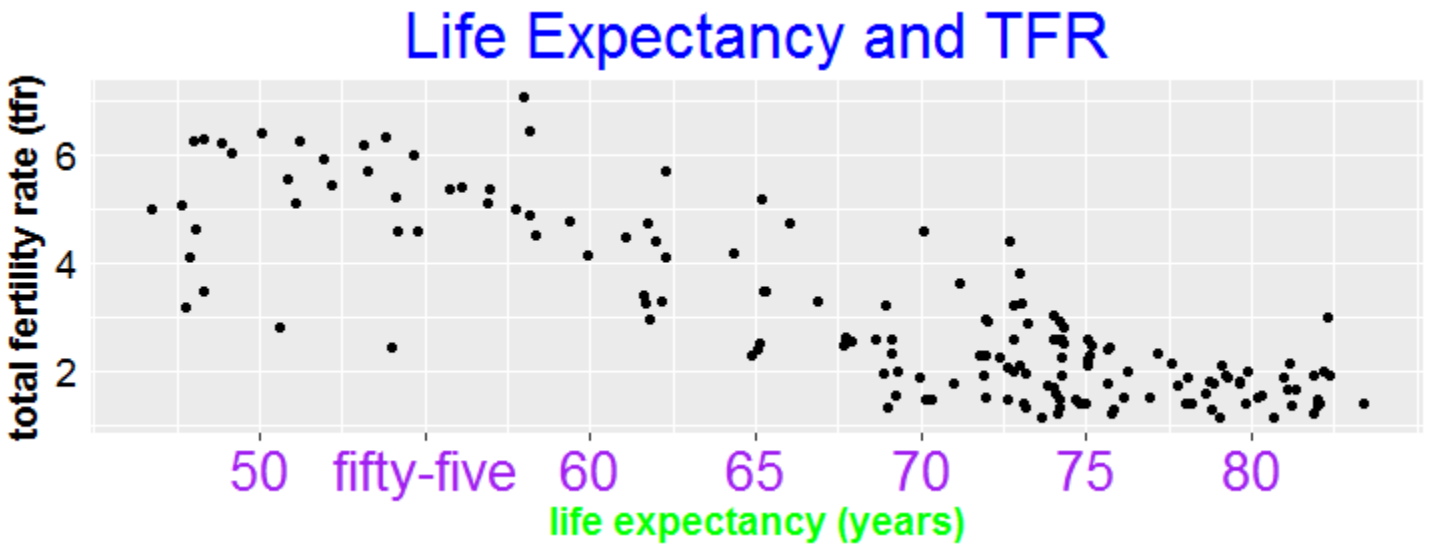
helps make plot visually pleasing by allowing addition/modification/deletion of titles, axis labels, tick marks, axis tick labels and legends

theme elements **inherit** properties from other theme elements, for example:



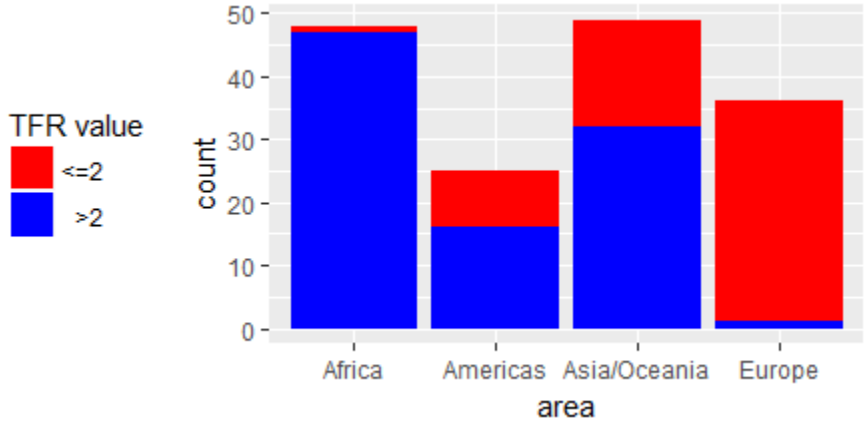
Theme: Titles, Tick Marks, and Tick Labels

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
p + geom_jitter() + ggtitle("Life Expectancy and TFR") +
xlab("life expectancy (years)") +
ylab("total fertility rate (tfr)") +
scale_x_continuous(breaks=seq(50,80,by=5),
                   labels=c(50,"fifty-five",60,65,70,75,80)) +
theme(plot.title=element_text(color="blue", size=24, hjust= 0.5),
      axis.title=element_text(size=14,face="bold"),
      axis.title.x=element_text(color="green"),
      axis.text=element_text(size=14),
      axis.text.y=element_text(color="black"),
      axis.text.x=element_text(color="purple", size=20),
      axis.ticks.y=element_blank())
```

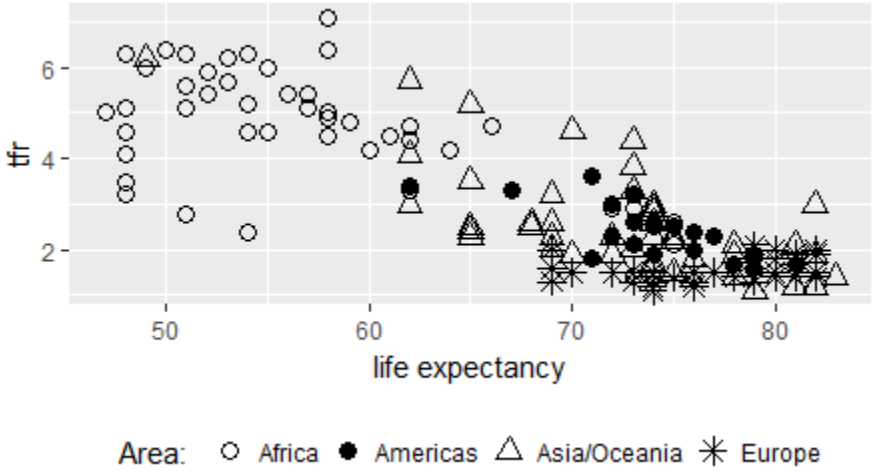


Theme: Legends

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w, aes(x=area, fill=tfrGT2))
```




```
p + geom_bar() +
  scale_fill_manual(name="TFR value",
        values = c("red","blue"),
        labels=c("<=2", ">2")) +
  theme(legend.position="left",
        legend.text.align=1)
```

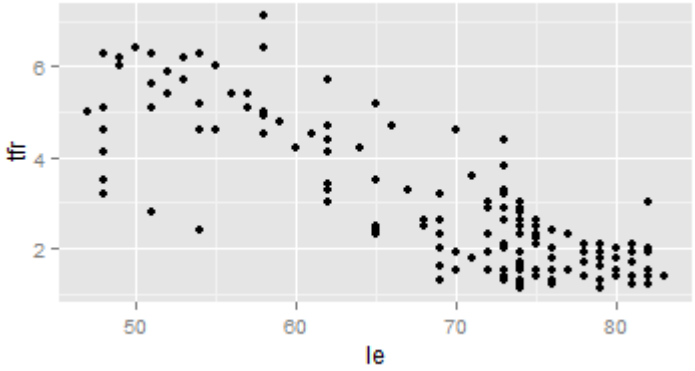



```
p + geom_point(aes(x=le, y=tfr,
    shape=area, fill=NULL), size = 3) +
  xlab("life expectancy") +
  scale_shape_manual(name="Area: ",
    values=c(1,16,2,8)) +
  theme(legend.key=element_blank(),
    legend.direction="horizontal",
    legend.position="bottom")
```

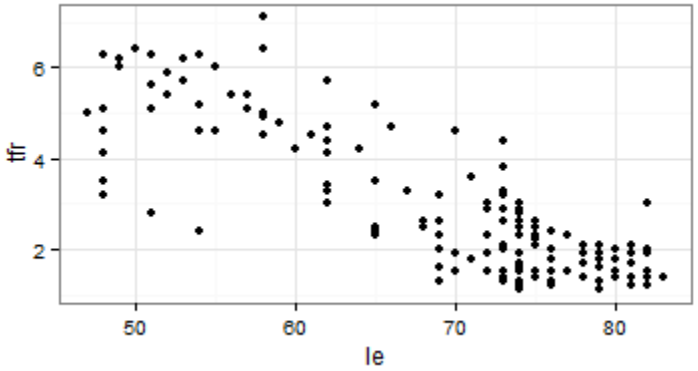
Theme: Overall Look

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr))
```

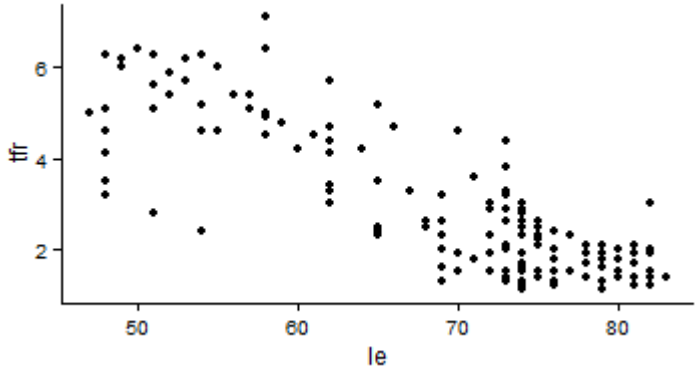
```
p + geom_point() + theme_gray()
```



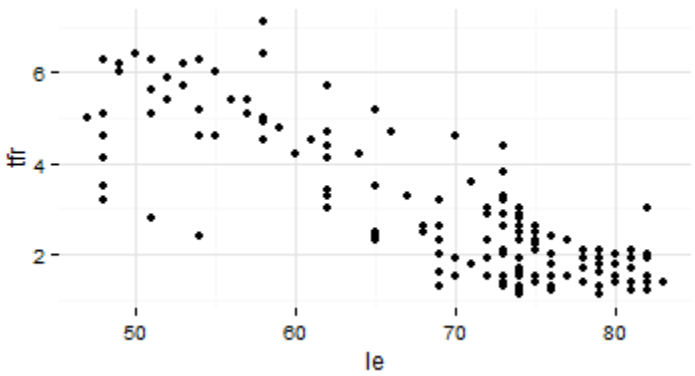

```
p + geom_point() + theme_bw()
```




```
p + geom_point() + theme_classic()
```




```
p + geom_point() + theme_minimal()
```

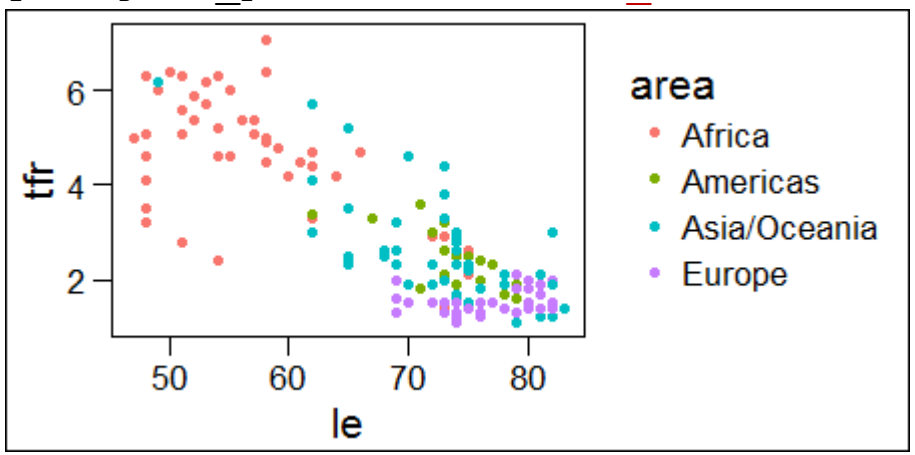


to change default theme use theme_set() ... for example, **`theme_set(theme_classic())`**

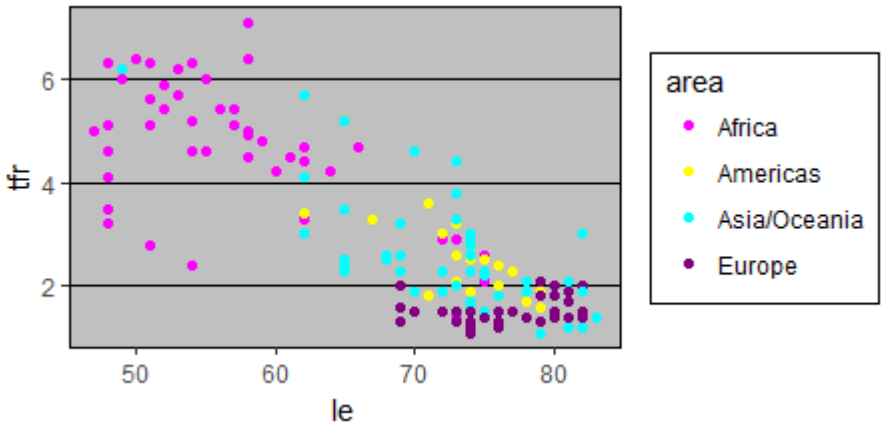
Themes - More Overall Looks

```
install.packages("ggthemes")
library("ggthemes")
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
```

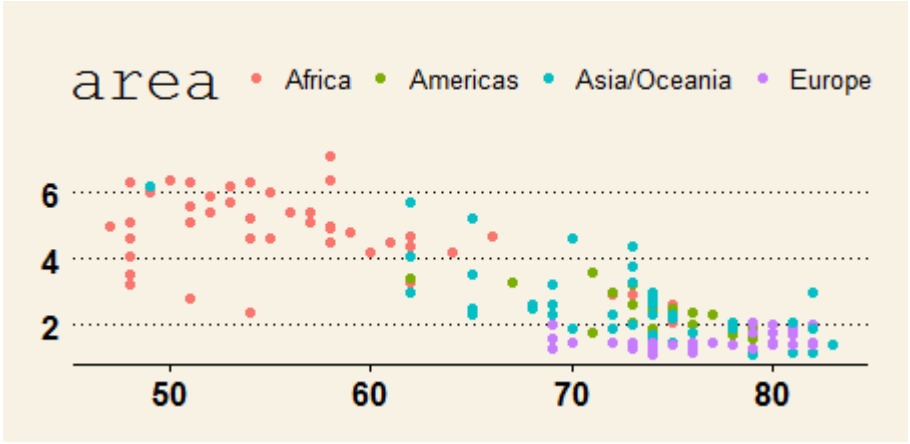
```
p + geom_point() + theme_base()
```



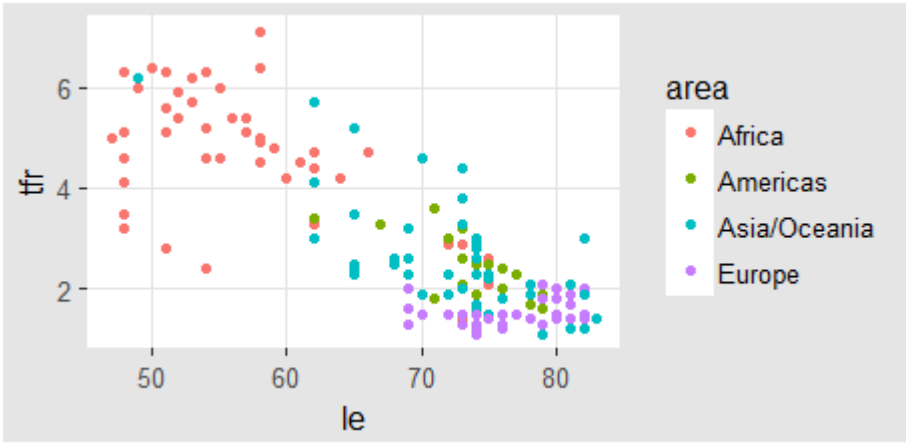

```
p + geom_point() + theme_excel()+
    scale_color_excel()
```




```
p + geom_point() + theme_wsj()
```



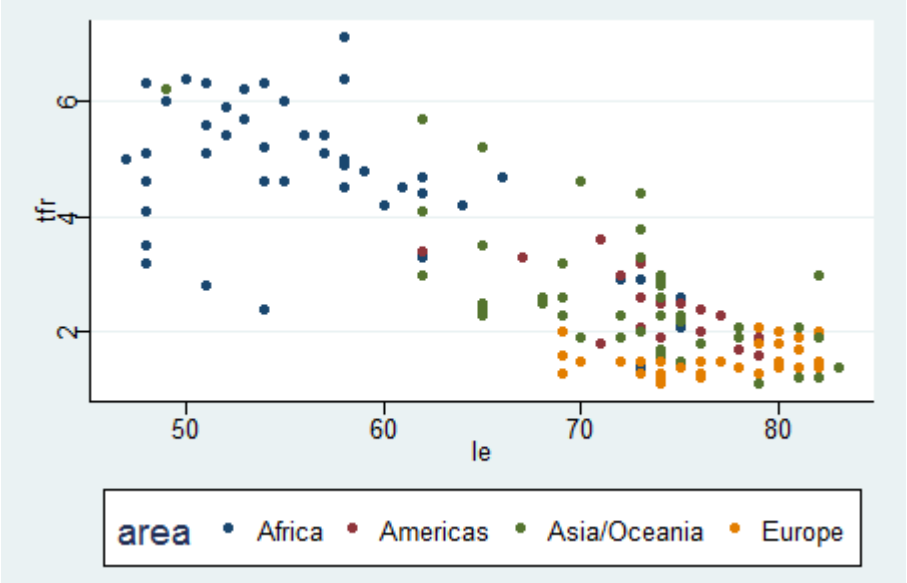

```
p + geom_point() + theme_igray()
```



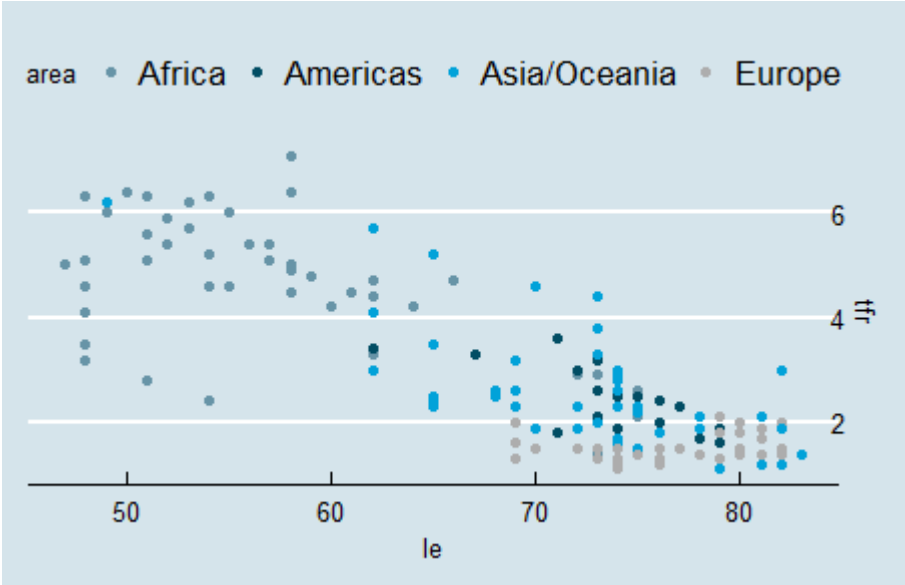
Themes - More Overall Looks

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
```

```
p + geom_point() + theme_stata() +
scale_color_stata()
```



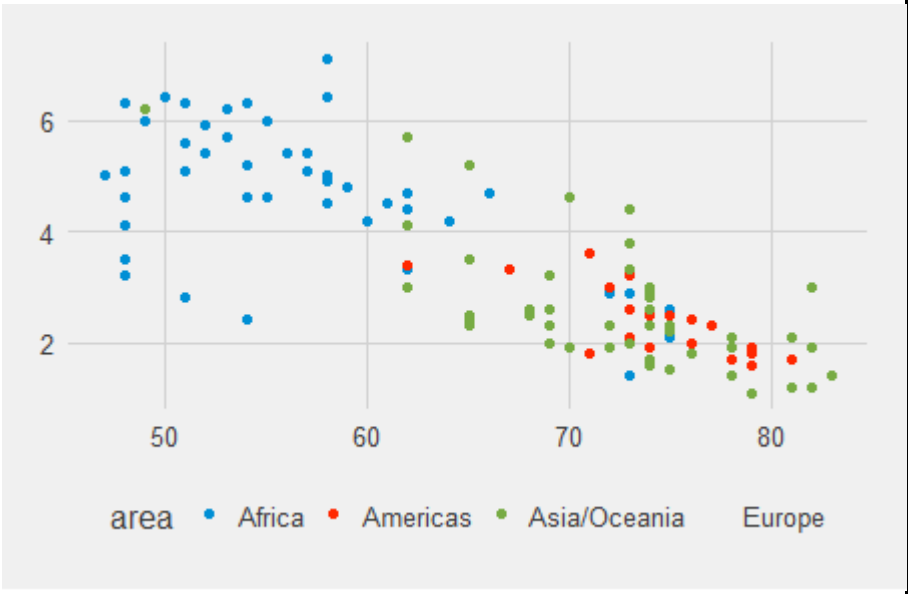

```
p + geom_point() +
theme_economist() +
scale_color_economist() +
scale_y_continuous(pos="right")
```



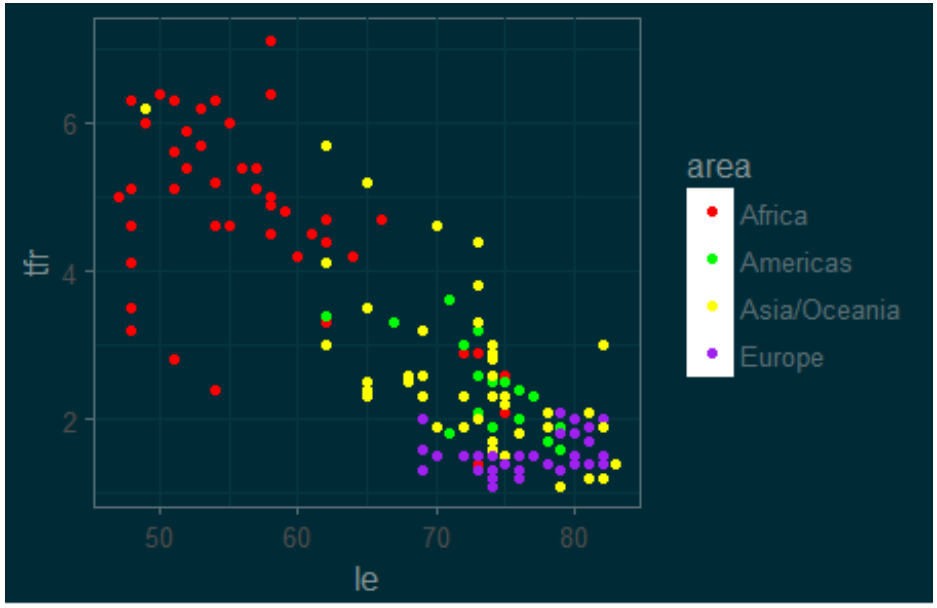
Themes - More Overall Looks

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=tfr, color=area))
```

```
p + geom_point() +
theme_fivethirtyeight() +
scale_color_fivethirtyeight
```




```
p + geom_point() +
theme_solarized(light=FALSE) +
scale_color_manual(values=c("red",
"green","yellow","purple"))
```



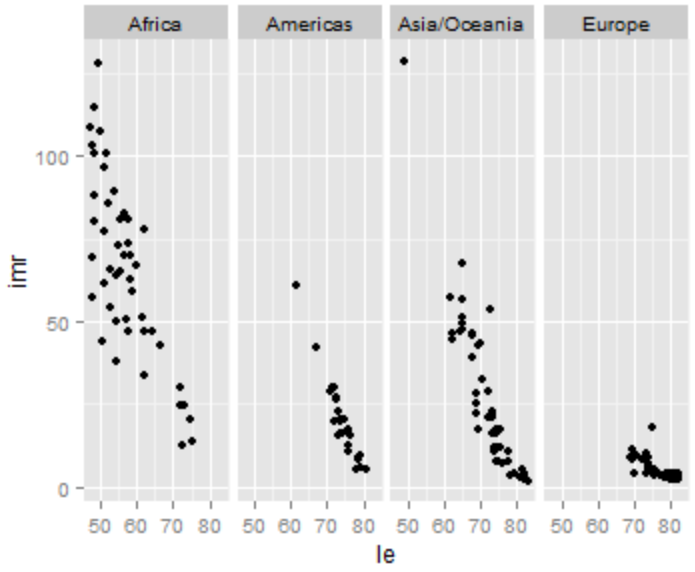
Facets

split data into subsets and plot each subset on a different panel

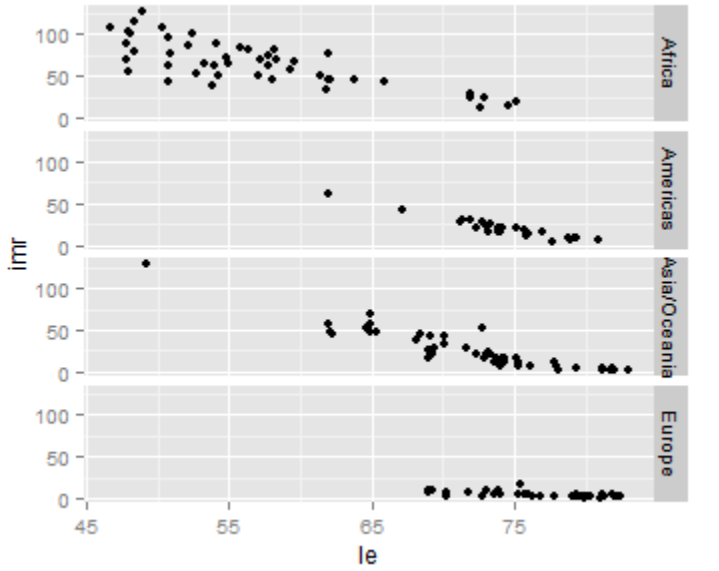
- show data as "small multiples"

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(data=w, aes(x=le, y=imr)) + geom_jitter()
```

```
p +  facet_grid(. ~ area)
```



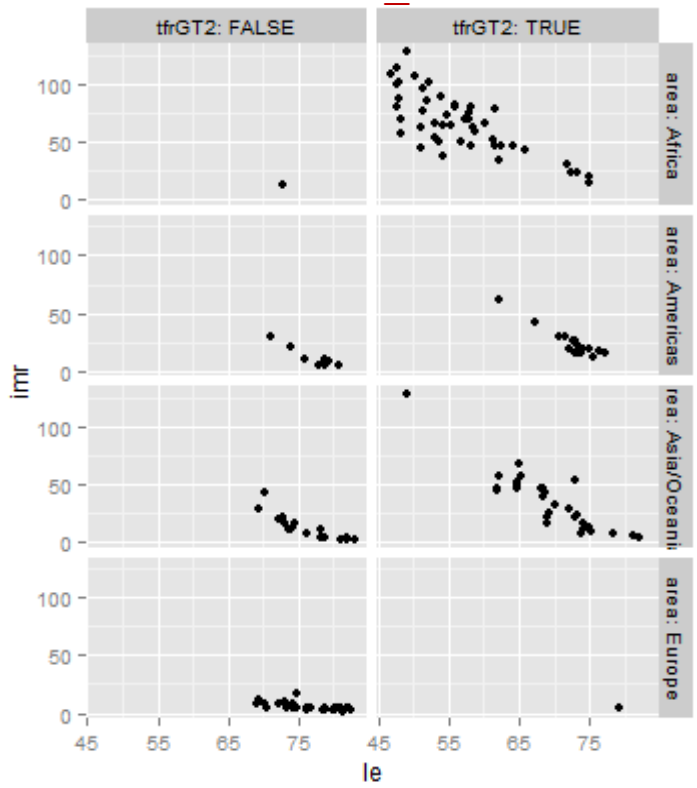

```
p + facet_grid(area ~ .)
```



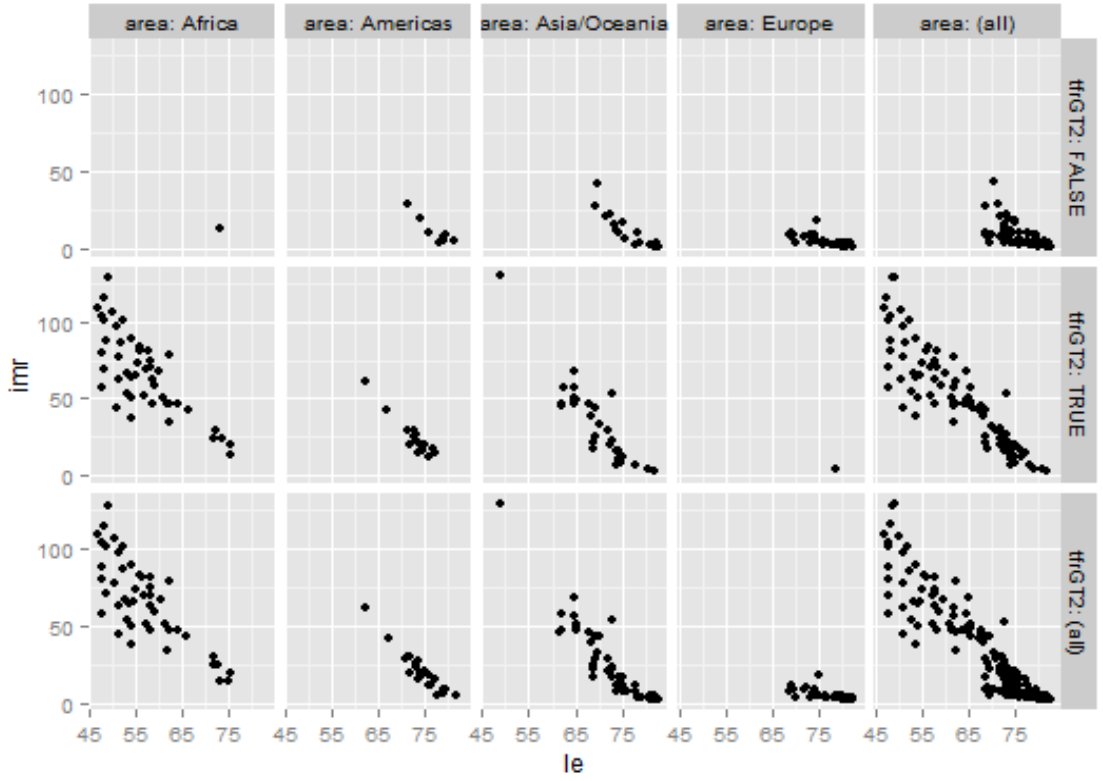
Facets

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w, aes(x=le, y=imr)) + geom_jitter()
```

```
p + facet_grid(area ~ tfrGT2,
labeller="label_both")
```




```
p + facet_grid(tfrGT2 ~ area,
labeller="label_both", margins=TRUE)
```



Saving Graphs

ggsave()

- saves last plot displayed
- requires file name to be supplied
- uses file name extension to determine file type:

.ps .eps .tex .pdf .jpg .tiff .png .bmp .svg .wmf (windows only)

- uses size of current graphics device for default size

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
ggplot(data=w, aes(x=le, y=tfr, color=area)) + geom_point()

ggsave(file="le_tfr1.jpg")
ggsave(file="le_tfr2.jpg", scale=2)
ggsave(file="le_tfr3.jpg", width=5, height=5, unit="in")

ggsave(file="le_tfr4.png")
ggsave(file="le_tfr5.pdf")
```

Part 2: Examples

Contents and Purpose of ggplot2 Graphs

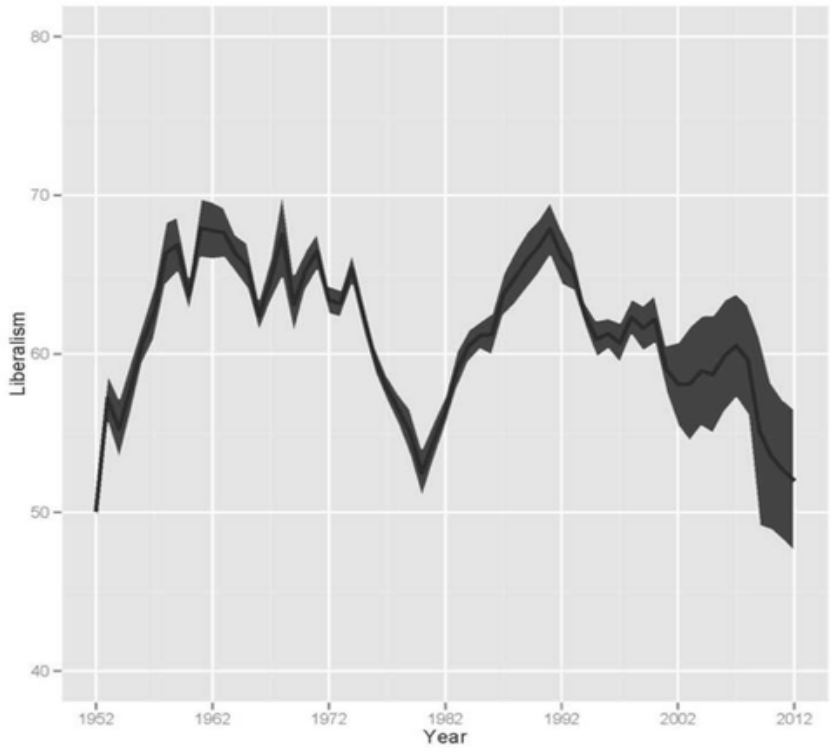
ggplot2 graph is typically created to show:

- data
- data + annotation
- statistical summary
- statistical summary + annotation
- data + statistical summary
- data + statistical summary + annotation

purpose of graph:

- **explore** data to
  increase understanding of data

- **communicate** about data ...
  often by showing data and/or
  statistical summary **plus** annotation

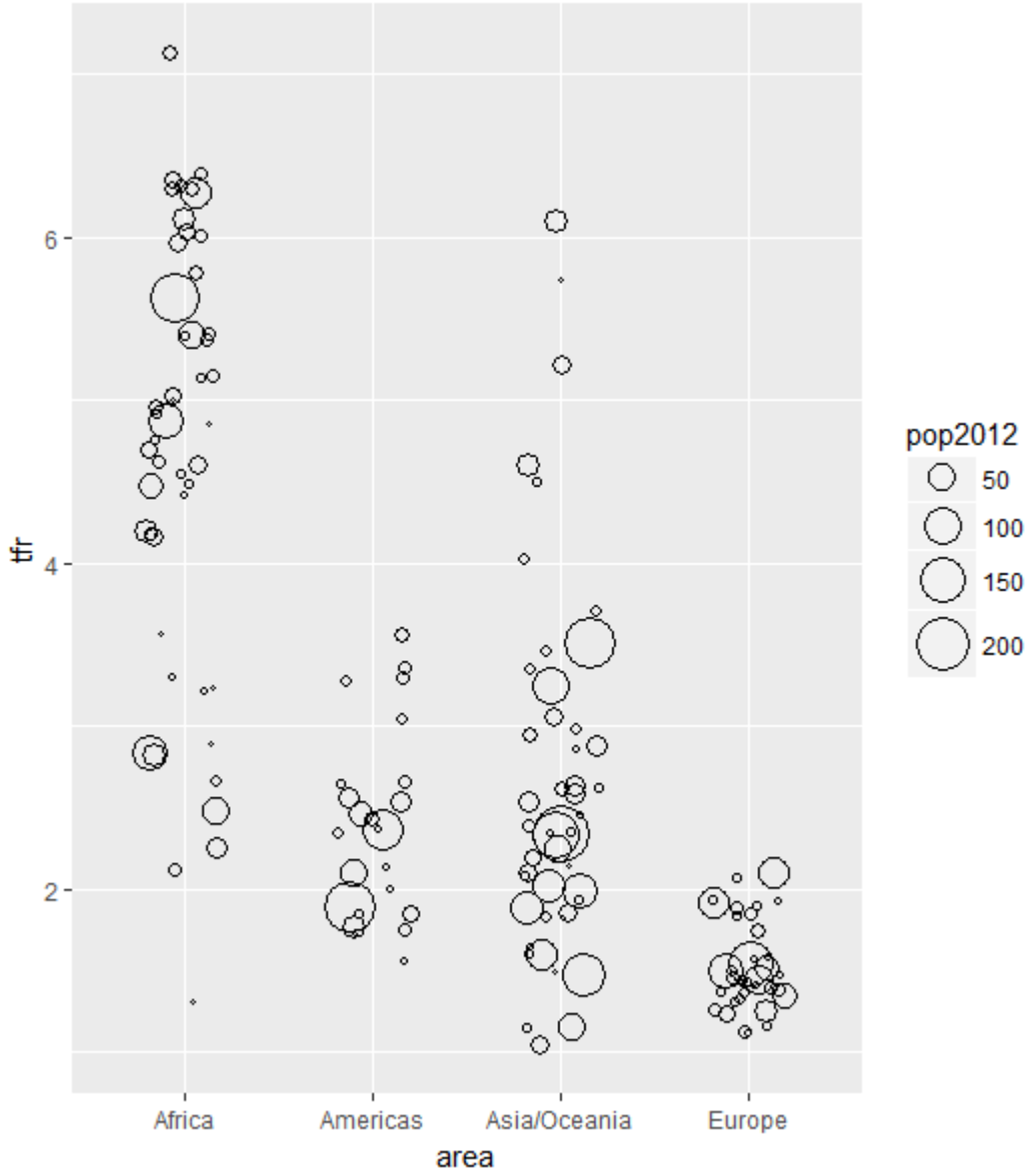


Graph associated with (online) NY Times Op-Ed piece by Thomas B. Edsall, "Does Rising Inequality Make Us Hardhearted?" December 10, 2013.

http://www.nytimes.com/imagepages/2013/12/11/opinion/11edsall-chart4.html?ref=opinion

Show Data

```
w <- read.csv(file="WDS2012.csv",
head=TRUE, sep=",")
popLT300 <- subset(w,pop2012<300)

p <- ggplot(data=popLT300,
     aes(x=area, y=tfr, size=pop2012)
p + geom_jitter(position=
position_jitter(w=.2, h=.1),shape=21)
scale_size_area(max_size=10)
```



Why is it important to show raw data?

Anscombe's Quartet

4 data sets that have
nearly identical summary statistics

each has 11 non-missing pairs of values

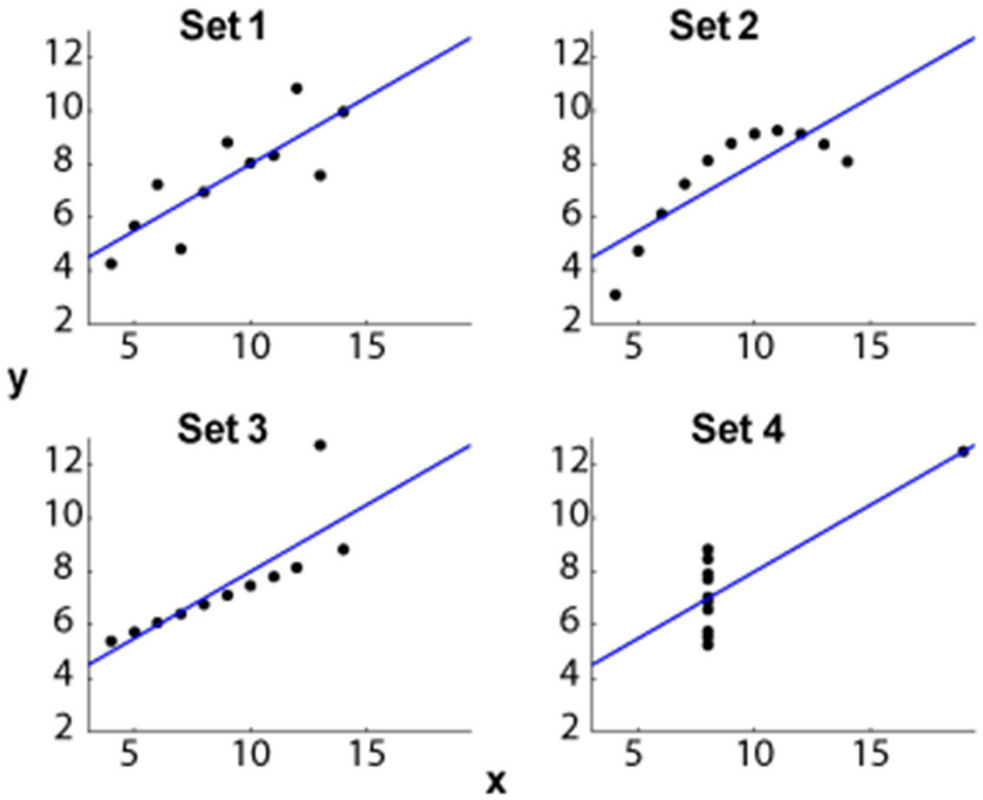
constructed in 1973 by statistician Francis Anscombe to demonstrate importance of graphing data and effect of outliers

Set 1		Set 2		Set 3		Set 4	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

SUMMARY STATISTICS				
mean value of x	9	9	9	9
mean value of y	7.5	7.5	7.5	7.5
variance of x	11	11	11	11
variance of y	4.1	4.1	4.1	4.1
correlation between x and y	0.816	0.816	0.816	0.816
linear regression (best fit) line is:	y=0.5x+3	y=0.5x+3	y=0.5x+3	y=0.5x+3

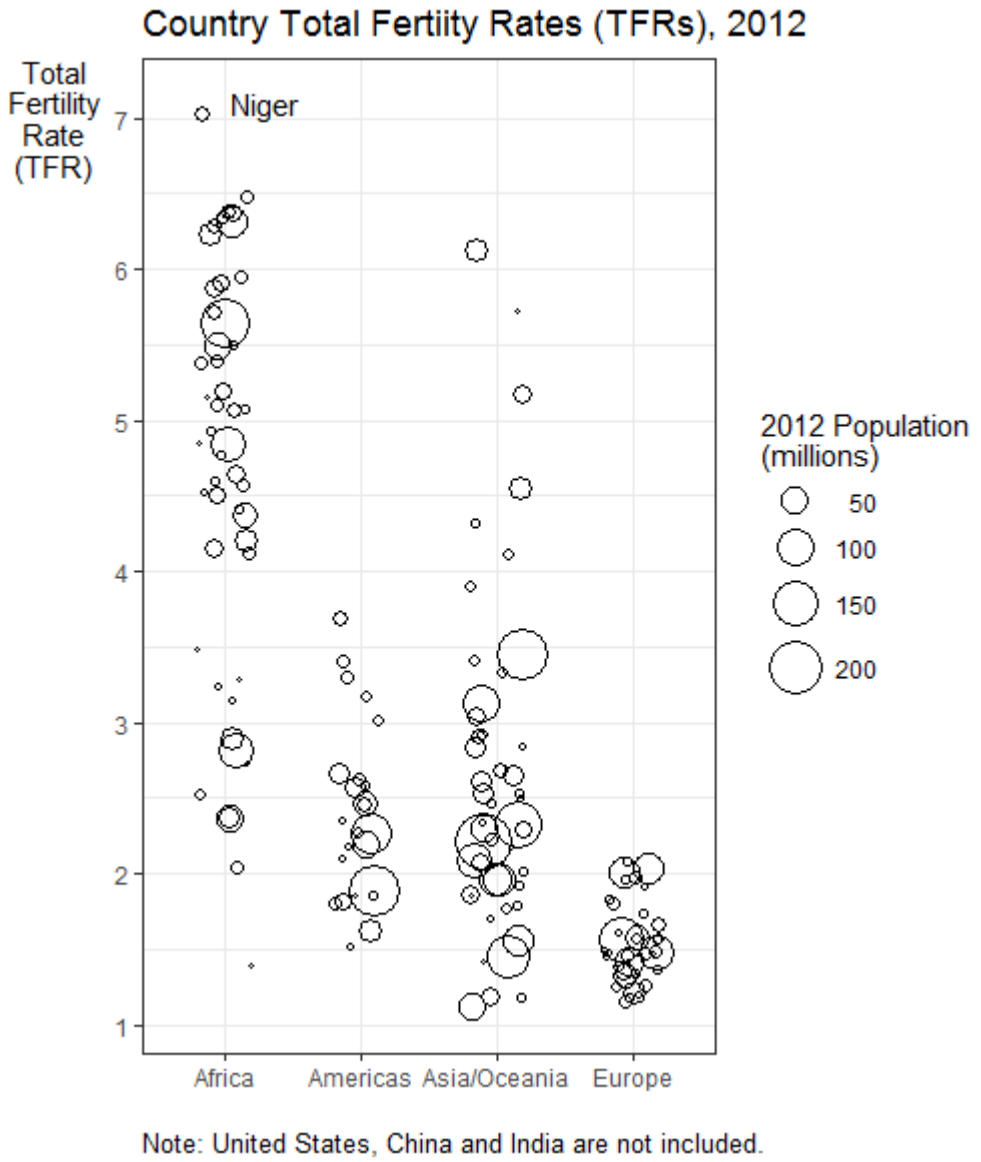
Anscombe, FJ (1973). "Graphs in Statistical Analysis". American Statistician 27 (1): 17–21.

Anscombe's Quartet



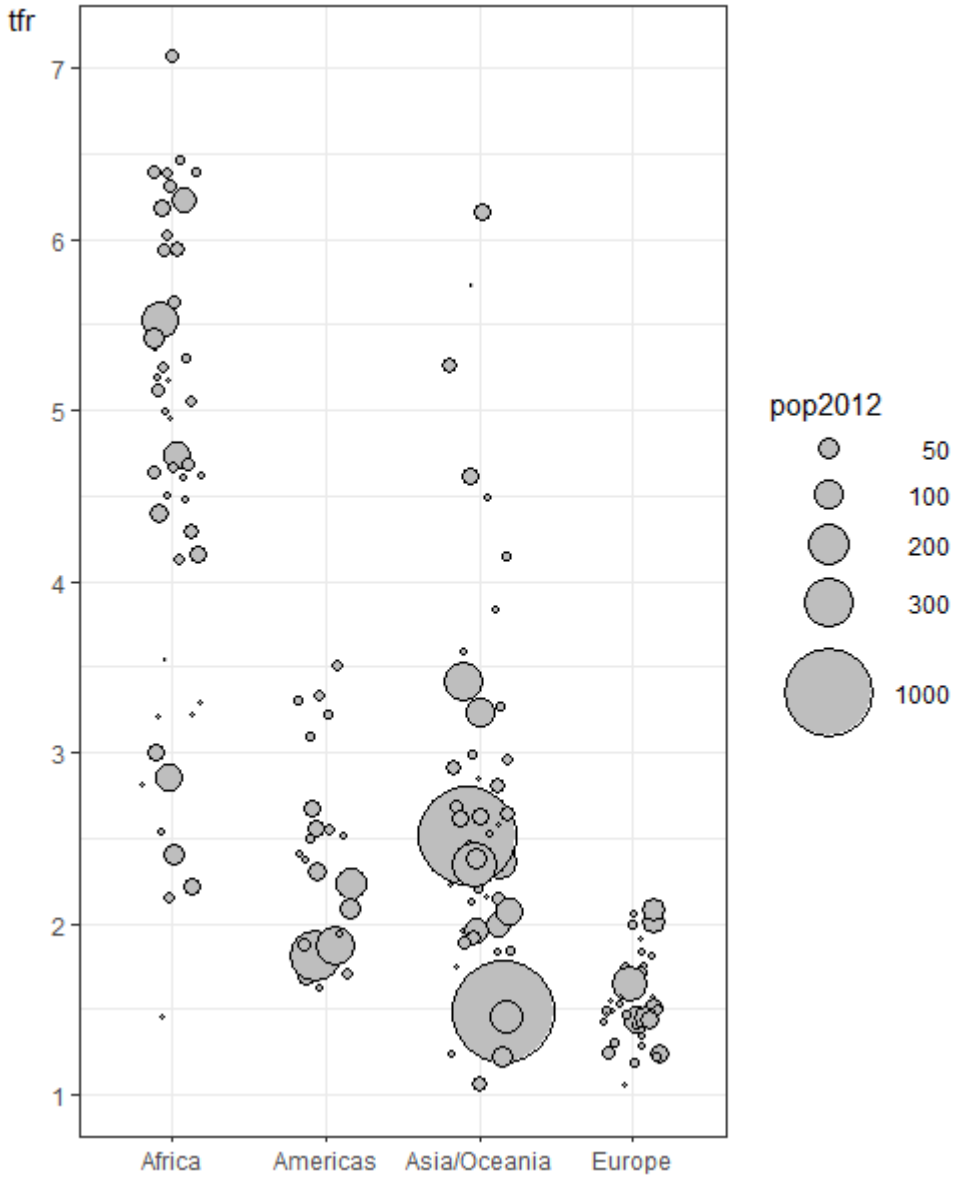
Data + Annotation

```
p <- ggplot(data=popLT300,
     aes(x=area, y=tfr, size=pop2012))
p + geom_jitter(position=
  position_jitter(w=.2, h=.1),shape=21)
scale_y_continuous(breaks=
  c(1,2,3,4,5,6,7)) +
scale_size_area(max_size=10) +
annotate("text", x=1.3,y=7.1,
         label="Niger", size=4) +
labs(title="Country Total Fertility Rate
     (TFRs), 2012",
     x="\nNote: United States, China and
     India are not included.",
     y="Total\nFertility\nRate\n(TFR)",
     size="2012 Population\n
      (millions)") +
theme_bw() +
theme(axis.title.x=element_text(size=10,
      hjust=0),
      axis.title.y=element_text(angle=0)
      legend.key=element_blank(),
      legend.text.align=1)
```



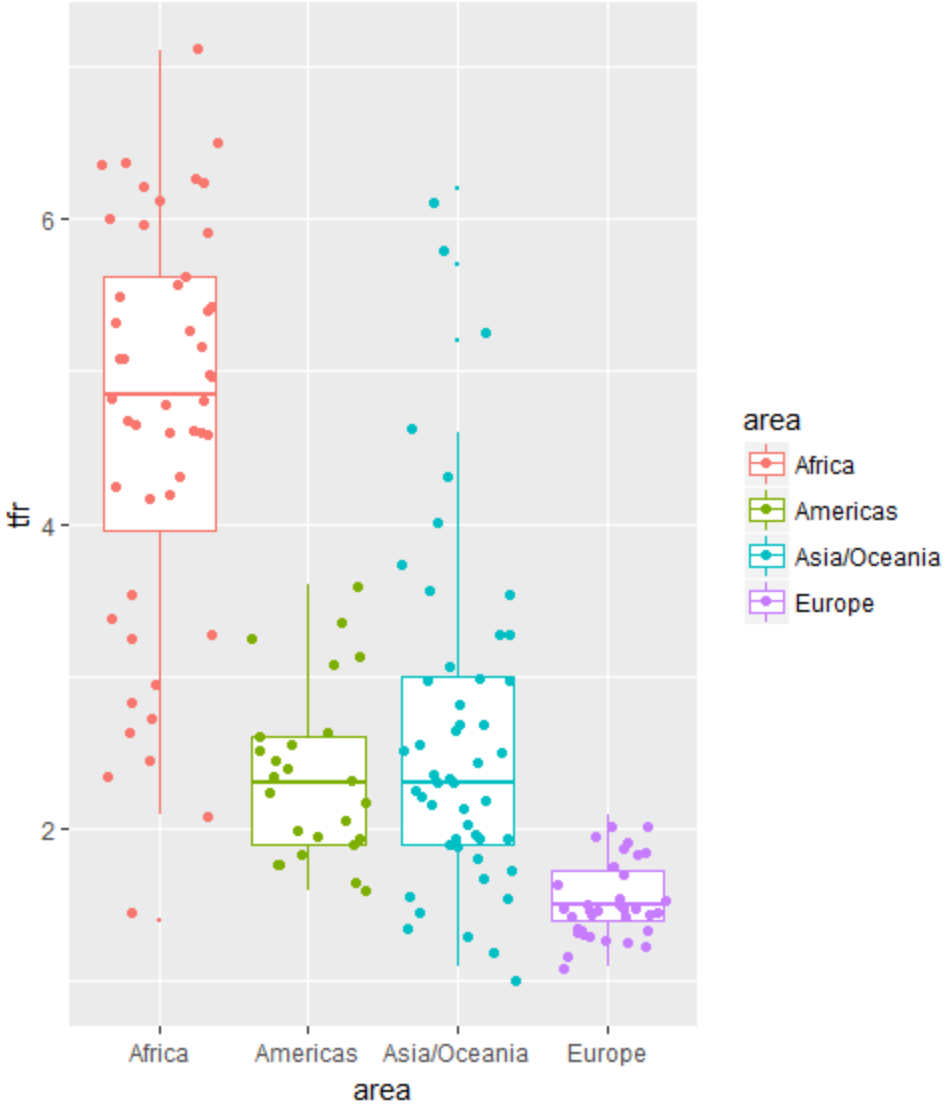
Show Data

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")

p <- ggplot(data=w, aes(x=area, y=tfr,
            size=pop2012))
p + geom_jitter(position=
    position_jitter(w=.2, h=.1),
    shape=21, fill="gray") +
scale_y_continuous(breaks=
    c(1,2,3,4,5,6,7)) +
scale_size_area(breaks=
    c(50,100,200,300,1000),
    max_size=18) +
theme_bw() +
theme(axis.title.x=element_blank(),
      axis.title.y=element_text(angle=0)
      legend.key=element_blank(),
      legend.text.align=1)
```



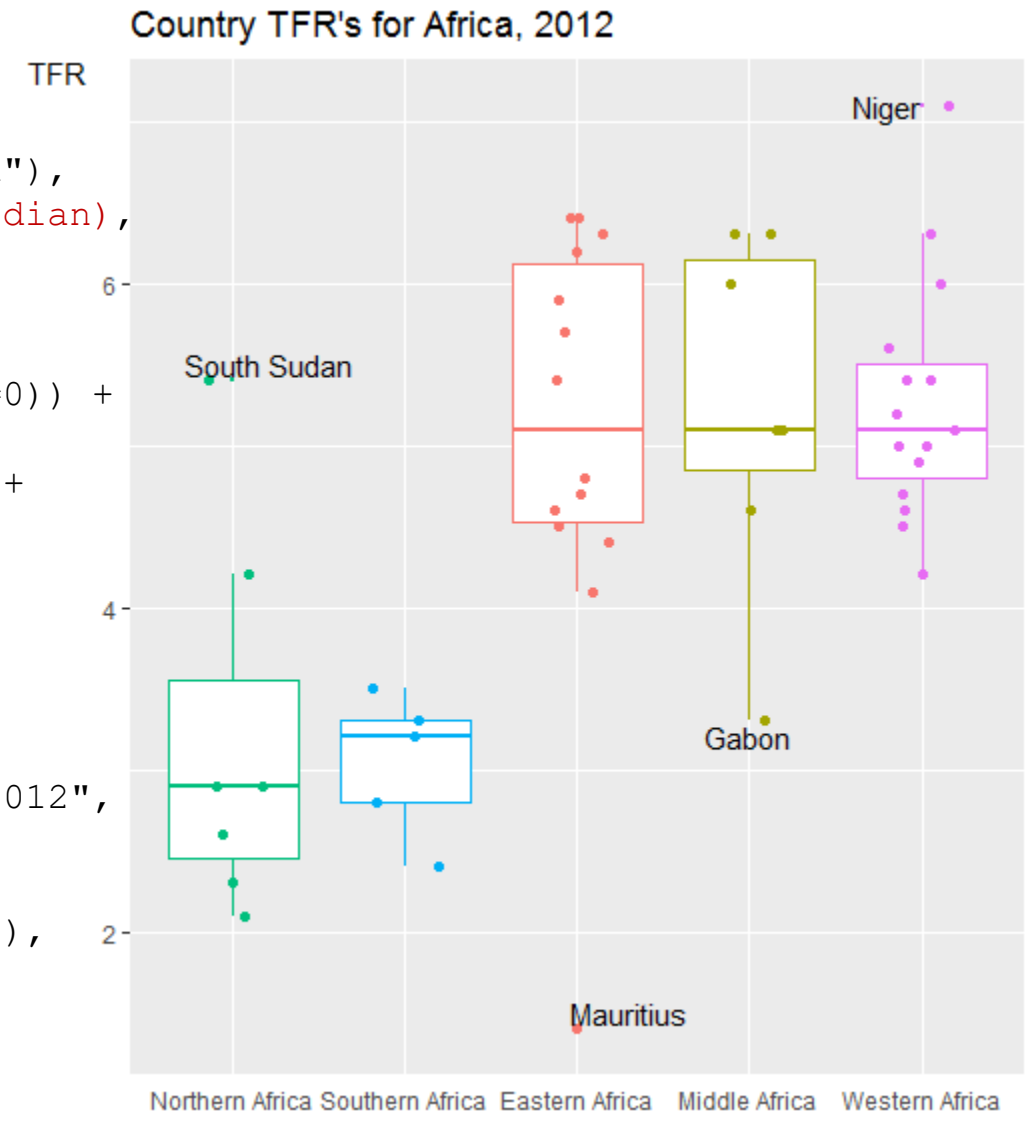
Data + Statistical Summary

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")

p <- ggplot(w, aes(x=area,
                   y=tfr,color=area))
p +  geom_boxplot(outlier.size=0) +
     geom_jitter(position=
                 position_jitter(h=.1))
```



Data + Statistical Summary + Annotation

```
p <- ggplot(data=subset(w,area=="Africa"),
aes(x=reorder(factor(region),tfr,FUN=median),
                 y=tfr, color=region))
p + geom_boxplot(outlier.size=0) +
    geom_jitter(position=
                 position_jitter(w=.2,h=0)) +
annotate("text",x=1.2, y=5.5,
          label="South Sudan", size=4) +
annotate("text",x=3.3, y=1.5,
          label="Mauritius", size=4) +
annotate("text",x=4.8, y=7.1,
          label="Niger", size=4) +
annotate("text",x=4, y=3.2,
          label="Gabon", size=4) +
labs(title="Country TFR's for Africa, 2012",
     x="", y="TFR") +
theme(axis.ticks.x=element_blank(),
      axis.title.y=element_text(angle=0),
      legend.position="none")
```



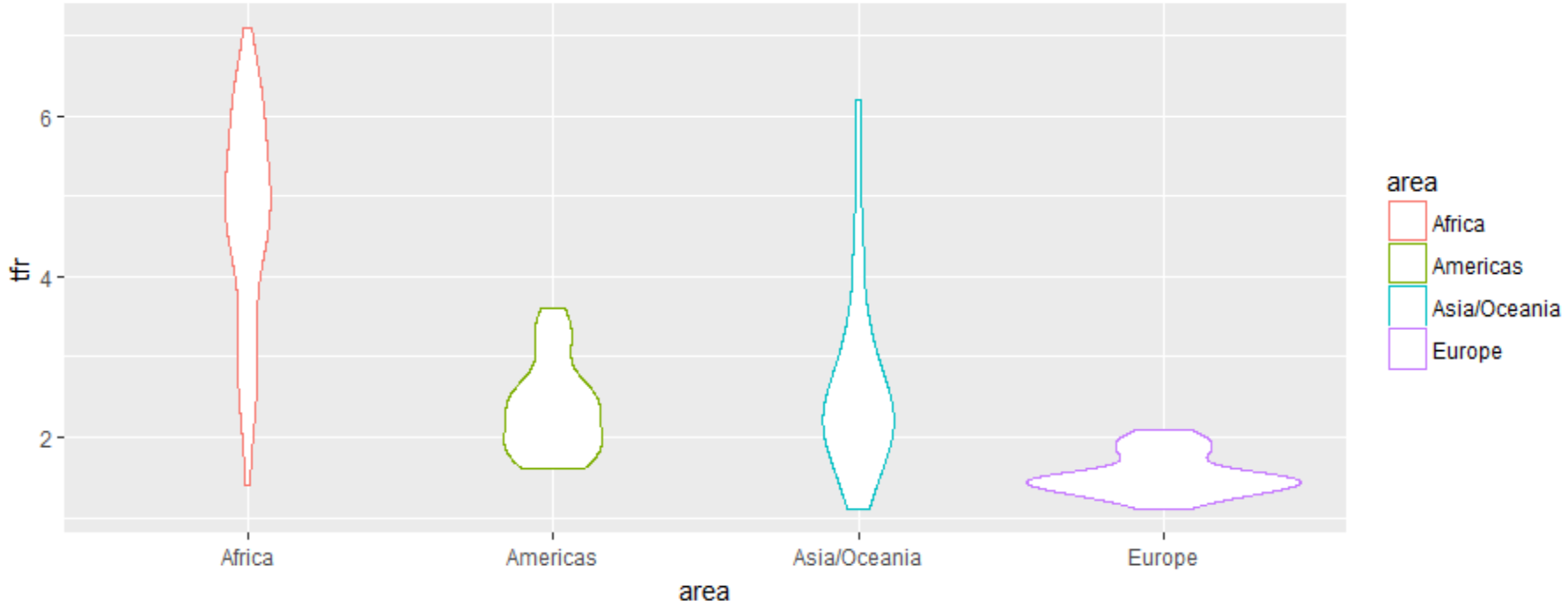
Statistical Summary

violin plot:

kernel density estimates, mirrored to have a symmetrical shape

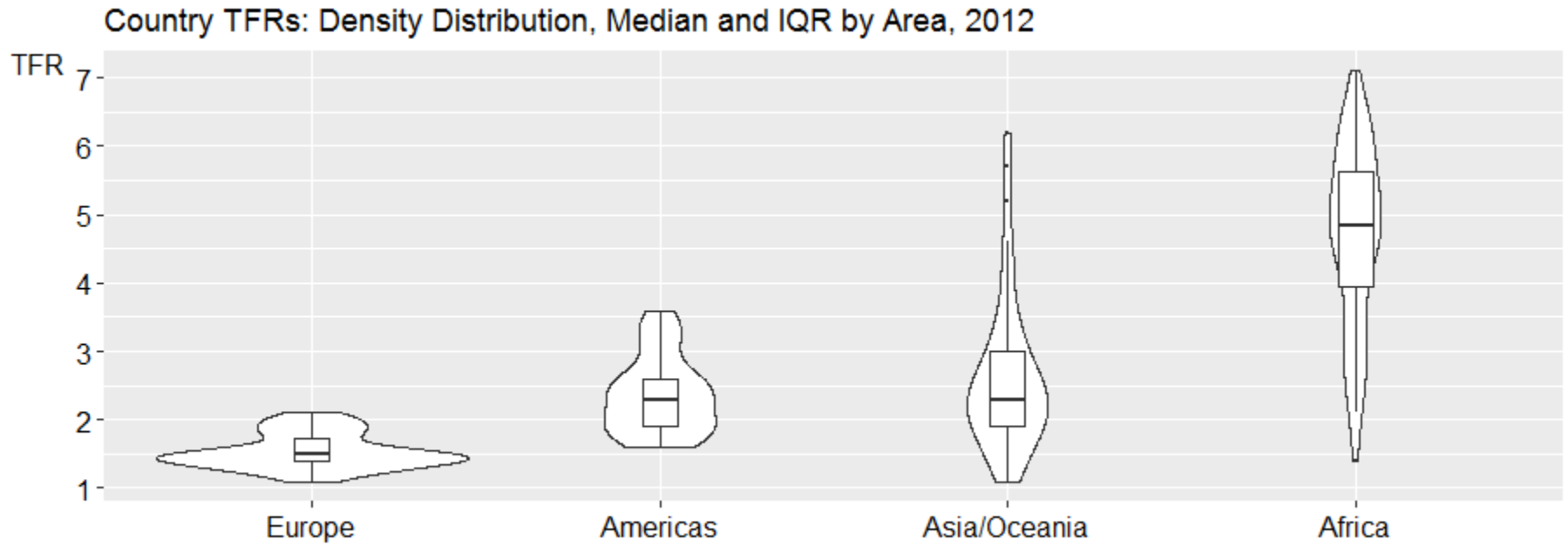
allows visual comparison of data distribution of several groups

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(w, aes(x=area, y=tfr, color=area))
p +  geom_violin()
```



Statistical Summaries

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")

p <- ggplot(w, aes(x=reorder(factor(area),tfr,FUN="median"), y=tfr))
p +  geom_violin() + geom_boxplot(width=.1, outlier.size=0) +
scale_y_continuous(breaks=c(1,2,3,4,5,6,7)) +
theme(axis.title.y=element_text(angle=0,size=12),
      axis.text.y=element_text(color="black", size=12),
      axis.text.x=element_text(color="black", size=12),
      legend.position="none") +
labs(title="Country TFRs: Density Distribution, Median and IQR by Area,
2012", x="", y="TFR")
```

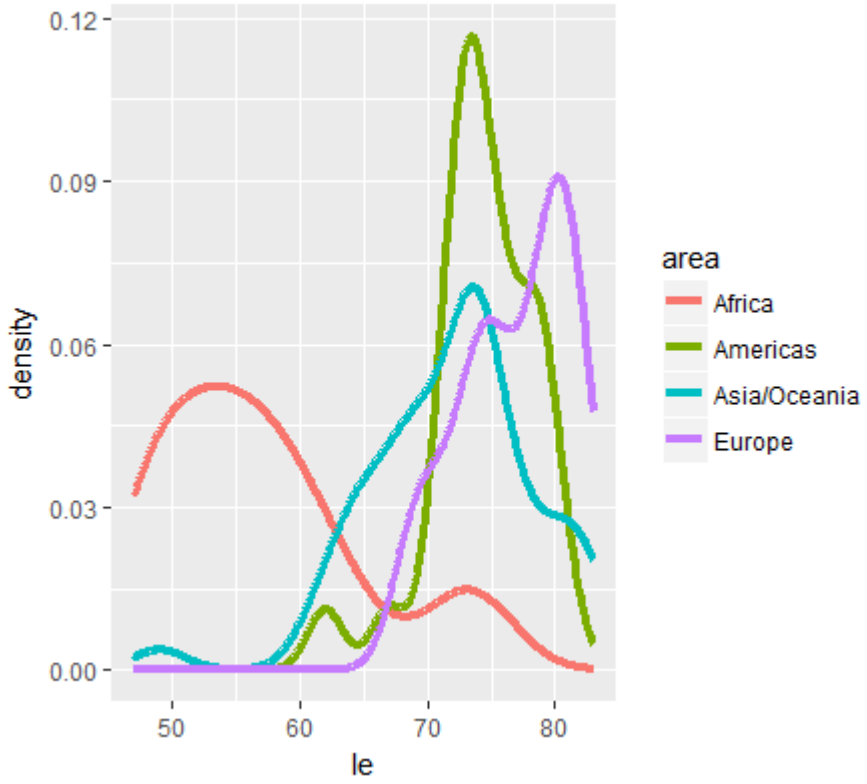


Statistical Summary

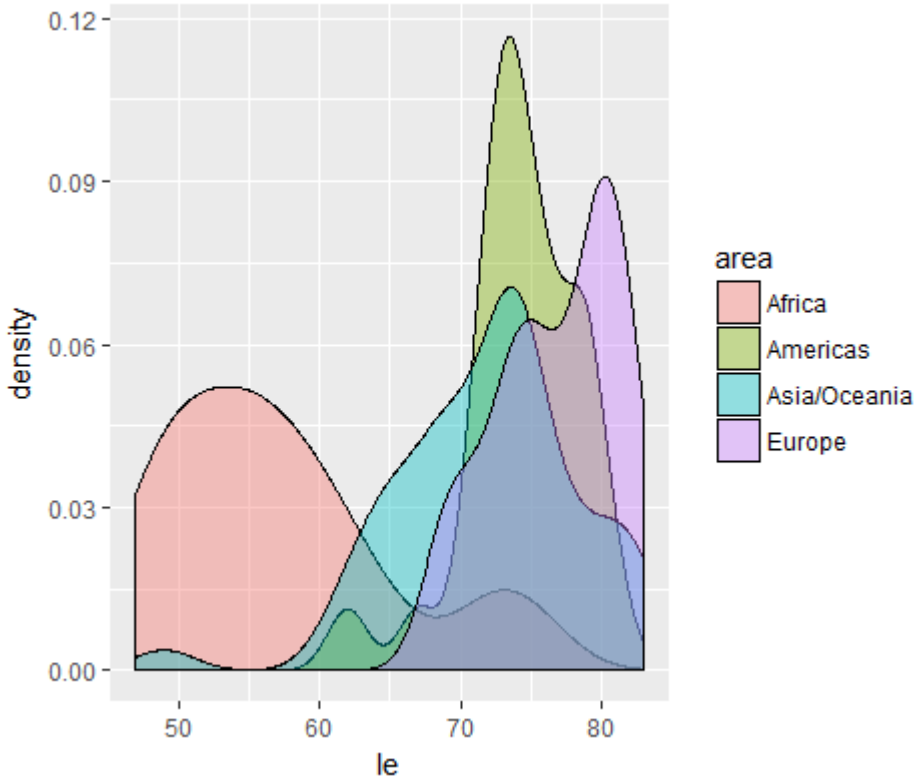
density distribution

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
```

```
p <- ggplot(w, aes(x=le, color=area))
p +  geom_line(stat="density", size=1.5)
```

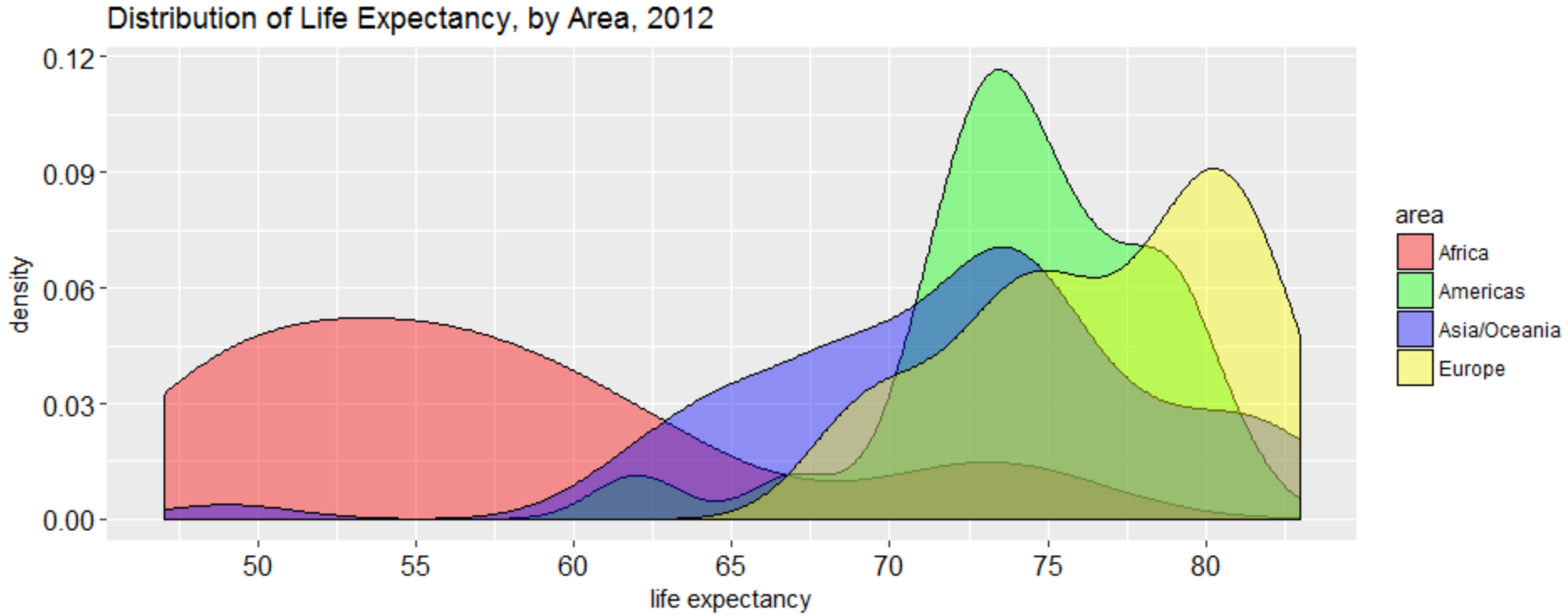



```
p <- ggplot(w, aes(x=le, fill=area))
p +  geom_density(alpha=.4)
```



Statistical Summary + Annotation

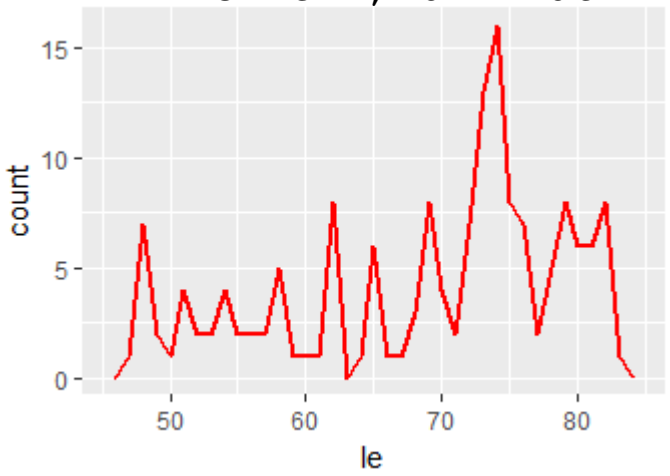
```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(w, aes(x=le, fill=area))
p +  geom_density(alpha=.4) +
  scale_fill_manual(values=c("red", "green", "blue", "yellow")) +
  scale_x_continuous(breaks=c(45,50,55,60,65,70,75,80,85)) +
  theme(axis.text=element_text(color="black", size=12)) +
  labs(title="Distribution of Life Expectancy, by Area, 2012", x="life expectancy")
```



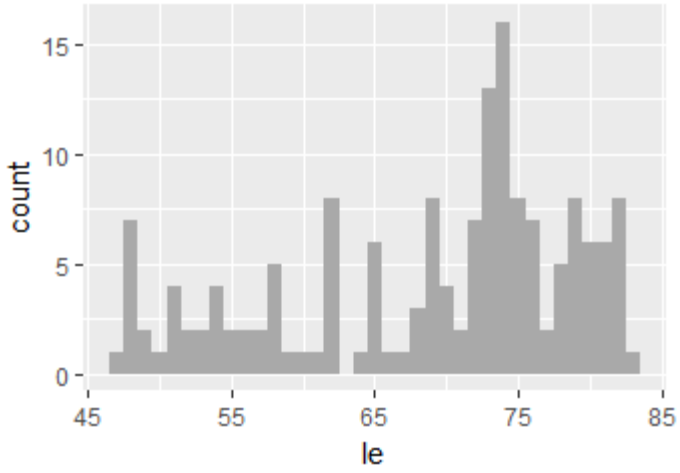
Statistical Summaries

```
w <- read.csv(file="WDS2012.csv", head=TRUE, sep=",")
p <- ggplot(w, aes(x=le))
```

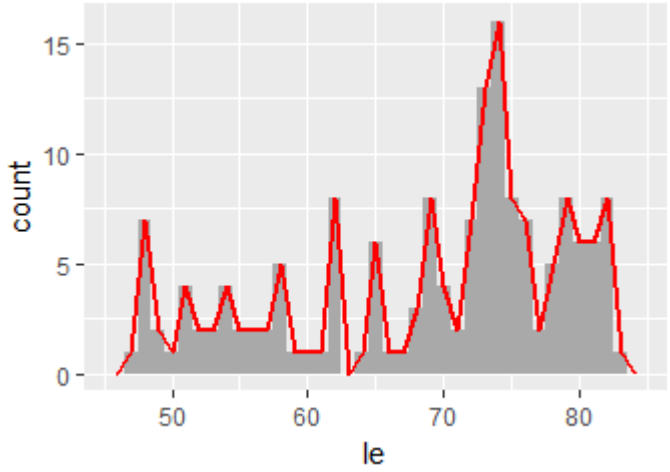
```
p + geom_freqpoly(color="red",
          size=1, binwidth=1)
```




```
p + geom_histogram(fill="darkgray",
                   binwidth=1)
```

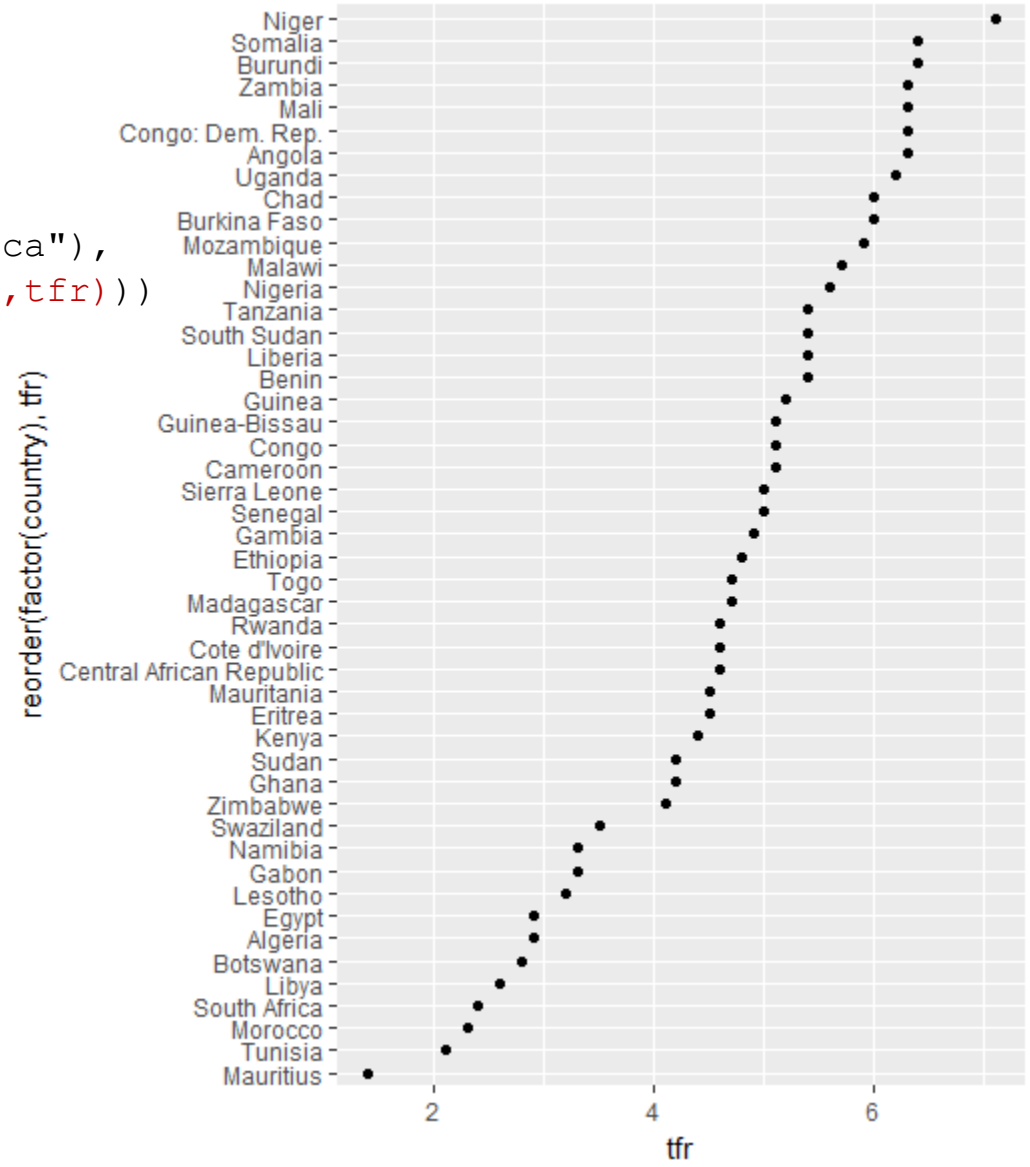



```
p + geom_histogram(fill="darkgray", binwidth=1) +
    geom_freqpoly(color="red", size=1, binwidth=1)
```



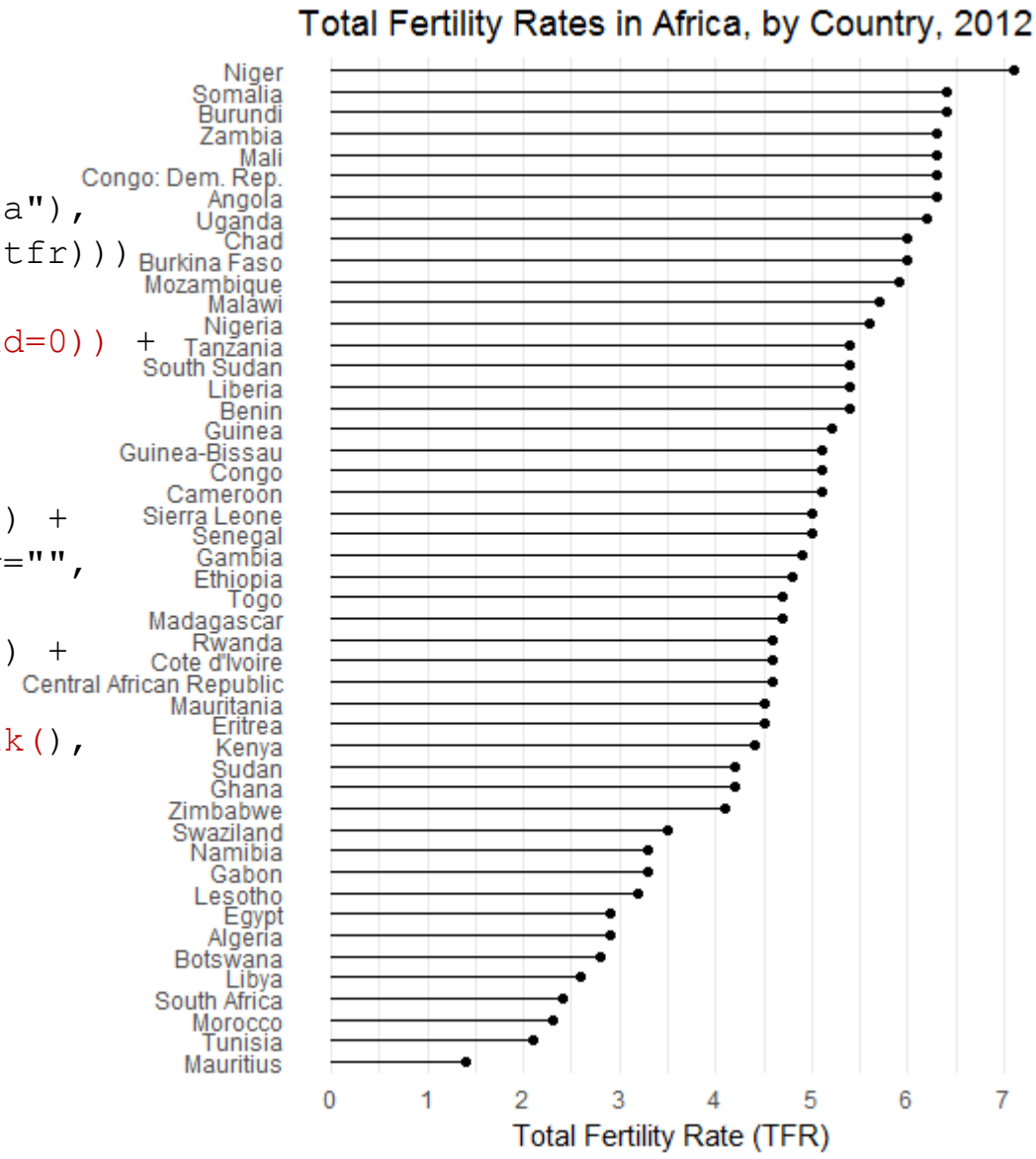
Show Data

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")

p <- ggplot(data=subset(w,area=="Africa"),
  aes(x=tfr,y=reorder(factor(country),tfr)))

p + geom_point()
```



Show Data

```
w <- read.csv(file="WDS2012.csv",
     head=TRUE, sep=",")
p <- gplot(data=subset(w,area=="Africa"),
aes(x=tfr,y=reorder(factor(country), tfr)))

p + geom_segment(aes(yend=country,xend=0)) +
geom_point() +
theme_minimal() +
scale_x_continuous(breaks=
                     c(0,1,2,3,4,5,6,7)) +
labs(x="Total Fertility Rate (TFR), y="",
     title="Total Fertility Rates
         in Africa, by Country, 2012") +

theme(panel.grid.major.y=element_blank(),
      axis.ticks=element_blank())
```

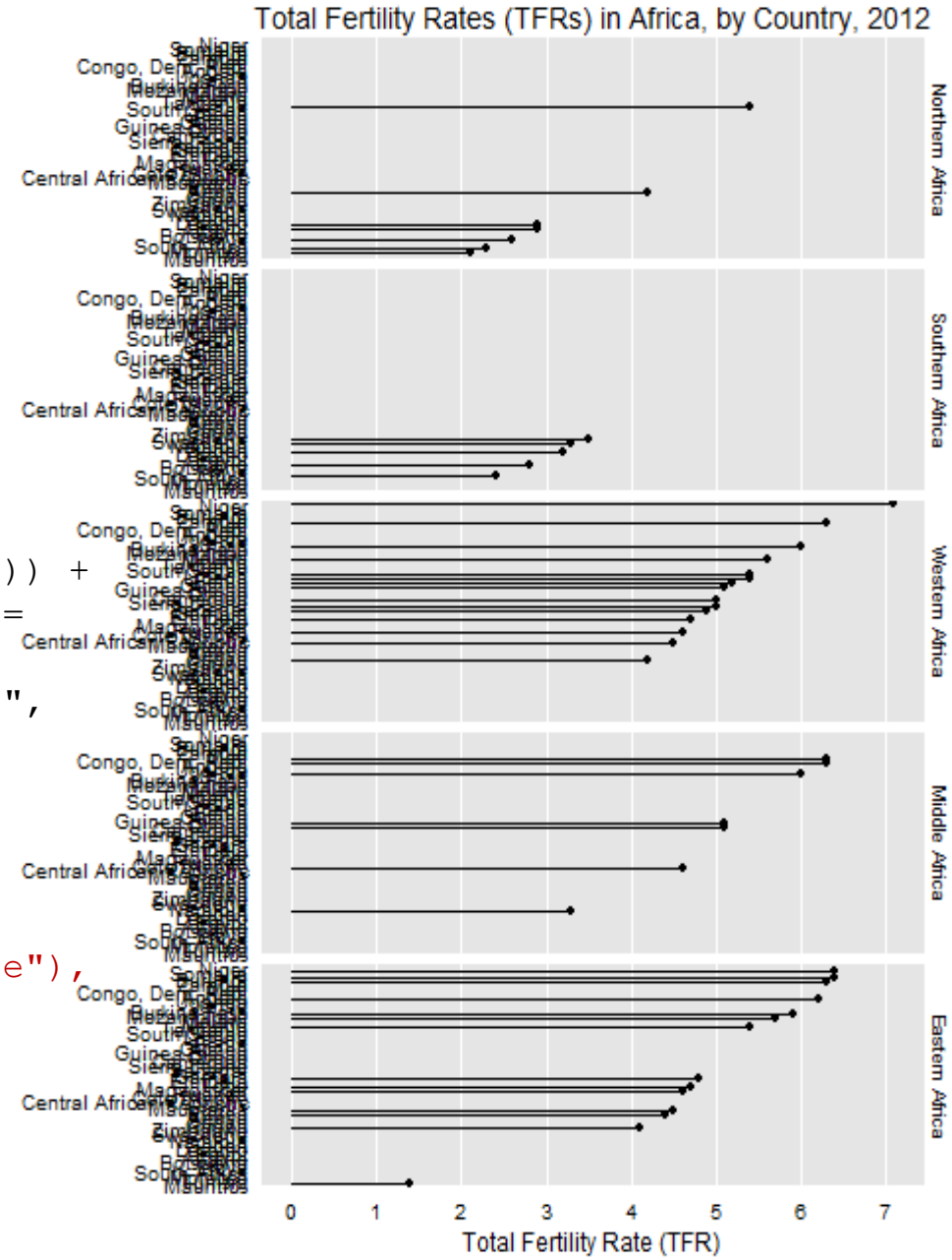


Show Data

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")

p <- ggplot(data=subset(w,area=="Africa"),
aes(x=tfr, y=reorder(factor(country),tfr)))

p + geom_text(aes(x=tfr-.1, label=country,
                  hjust=1), size=4) +
geom_point() +
theme_minimal() +
scale_x_continuous(breaks=c(1,2,3,4,5,6,7),
                   limits=c(0,8)) +
labs(x="", y="",
     title="Total Fertility Rates (TFRs) in
      Africa, by Country, 2012") +
theme(panel.grid.major.y=element_blank(),
      axis.text.y=element_blank(),
      axis.ticks=element_blank())
```

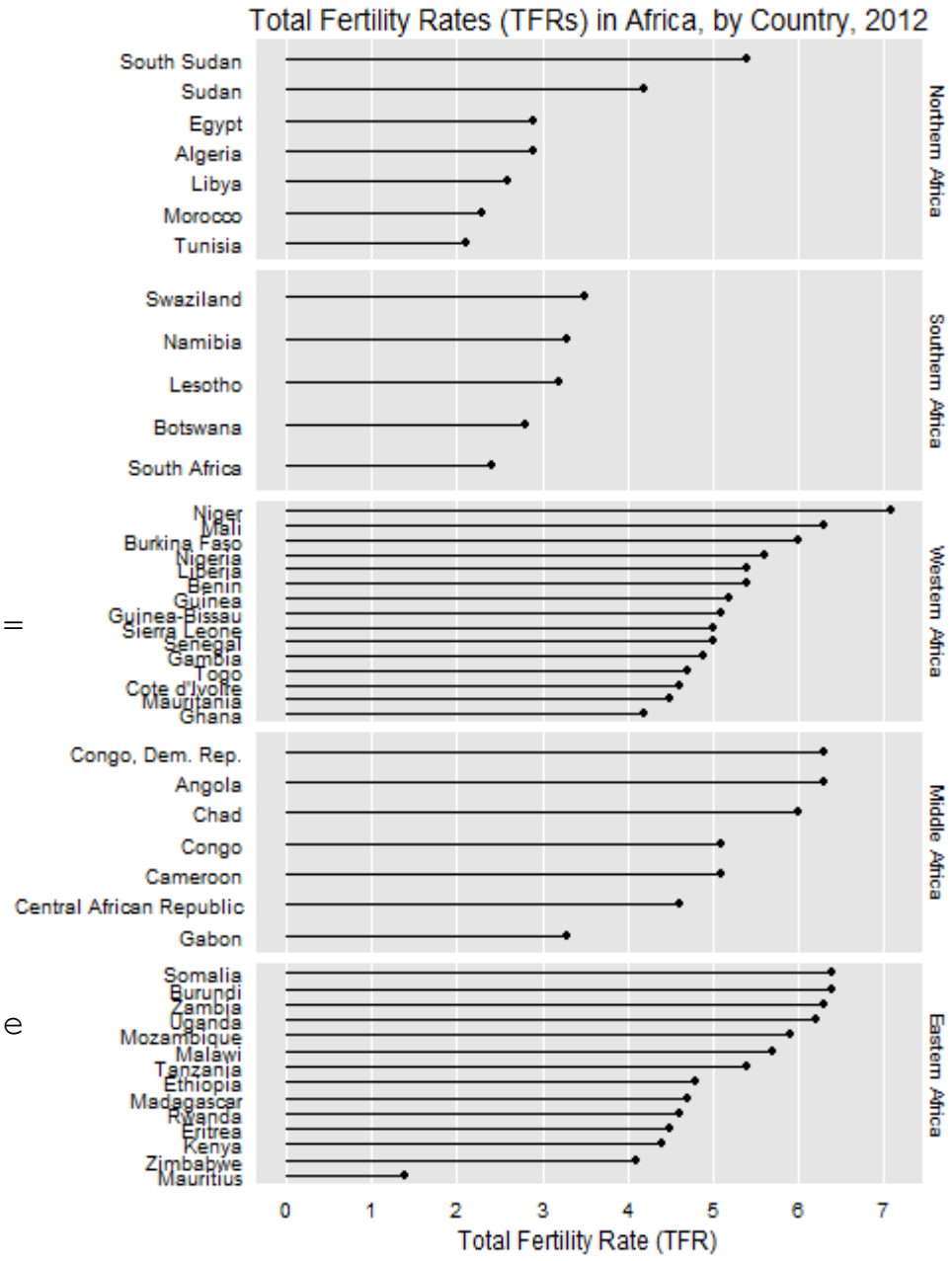
Show Data

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")
a <- subset(w,area=="Africa")
a$region <- factor(a$region,levels=
c("Northern Africa","Southern Africa",
"Western Africa", "Middle Africa",
"Eastern Africa" ))
p <- ggplot(data=a,aes(x=tfr,
     y=reorder(factor(country),tfr)))
p + geom_segment(aes(yend=country,xend=0)) +
geom_point() + scale_x_continuous(breaks=
               c(0,1,2,3,4,5,6,7)) +
labs(x="Total Fertility Rate (TFR)", y="",
title="Total Fertility Rates (TFRs) in
       Africa, by Country, 2012") +
theme(
axis.text=element_text(color="black"),
strip.text.y=element_text(size=9),
strip.background=element_rect(fill="white"),
panel.grid.major.y=element_blank(),
panel.grid.minor.x=element_blank(),
axis.ticks=element_blank()) +
facet_grid(region ~ .)
```



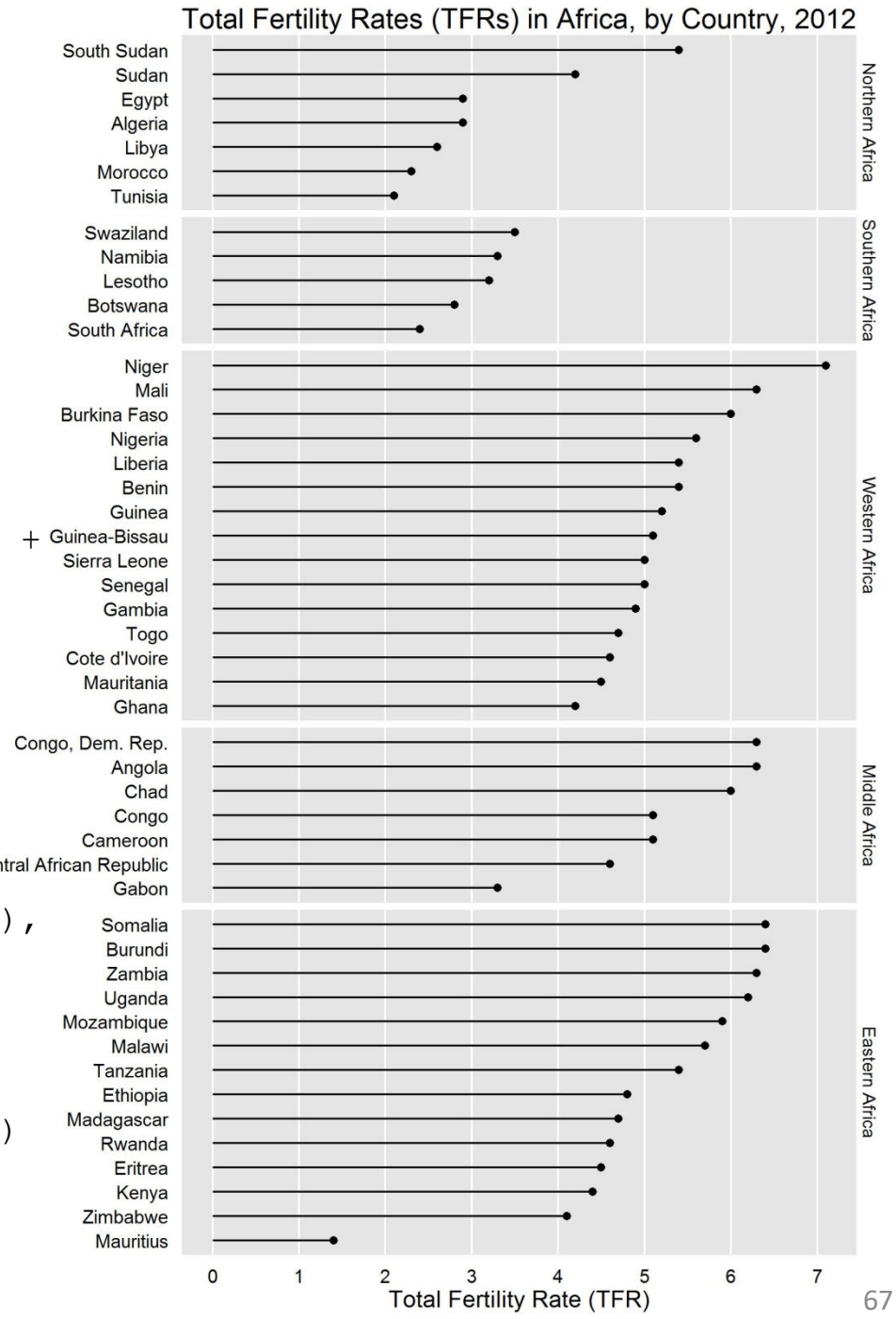
Show Data

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")
a <- subset(w,area=="Africa")
a$region <- factor(a$region,levels=
c("Northern Africa","Southern Africa",
"Western Africa", "Middle Africa",
"Eastern Africa" ))
p <- ggplot(data=a,aes(x=tfr,
     y=reorder(factor(country),tfr)))
p +
geom_segment(aes(yend=country,xend=0)) +
geom_point() + scale_x_continuous(breaks=
               c(0,1,2,3,4,5,6,7)) +
labs(x="Total Fertility Rate (TFR)",
y="",
title="Total Fertility Rates (TFRs) in
       Africa, by Country, 2012") +
theme(
axis.text=element_text(color="black"),
strip.text.y=element_text(size=9),
strip.background=element_rect(fill="white
"),
panel.grid.major.y=element_blank(),
panel.grid.minor.x=element_blank(),
axis.ticks=element_blank()) +
facet_grid(region ~ ., scales="free_y")
```



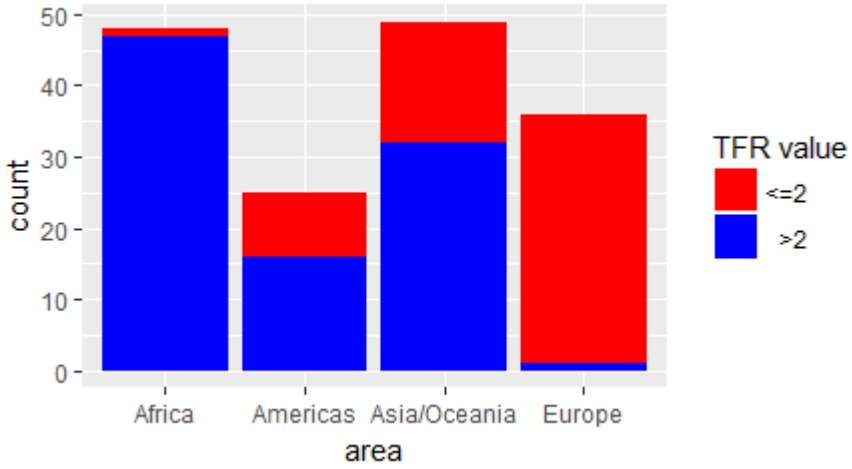
Show Data

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")
a <- subset(w,area=="Africa")
a$region <- factor(a$region,levels=
c("Northern Africa","Southern Africa",
"Western Africa", "Middle Africa",
"Eastern Africa" ))
p <- ggplot(data=a,aes(x=tfr,
     y=reorder(factor(country),tfr)))
p + geom_segment(aes(yend=country,xend=0)) +
geom_point() + scale_x_continuous(breaks=
               c(0,1,2,3,4,5,6,7)) +
labs(x="Total Fertility Rate (TFR)", y="",
title="Total Fertility Rates (TFRs) in
       Africa, by Country, 2012") +
theme(
axis.text=element_text(color="black"),
strip.text.y=element_text(size=9),
strip.background=element_rect(fill="white"),
panel.grid.major.y=element_blank(),
panel.grid.minor.x=element_blank(),
axis.ticks=element_blank()) +
facet_grid(region ~ .,
           scales="free_y", space="free_y")
```

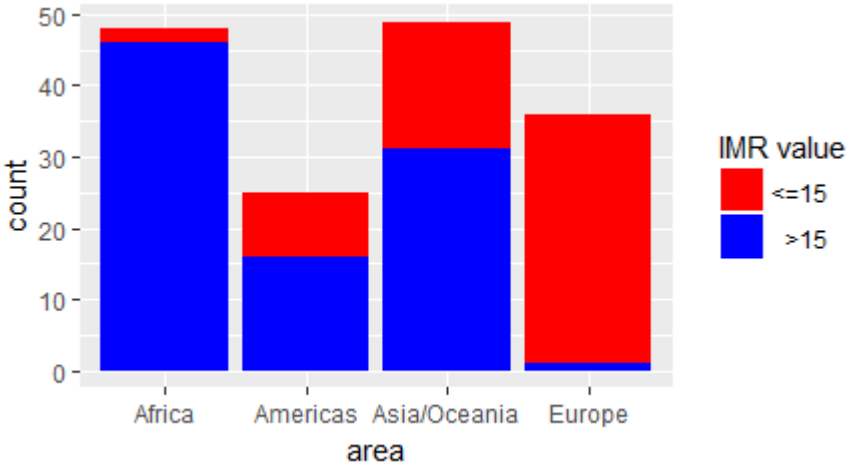


Statistical Summary

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")
w$tfrGT2 <- w$tfr > 2
p <- ggplot(data=w,
            aes(x=area, fill=tfrGT2))
p + geom_bar() +
scale_fill_manual(name="TFR value",
values = c("red","blue"),
labels=c("<=2", ">2")) +
theme(legend.text.align=1)
```

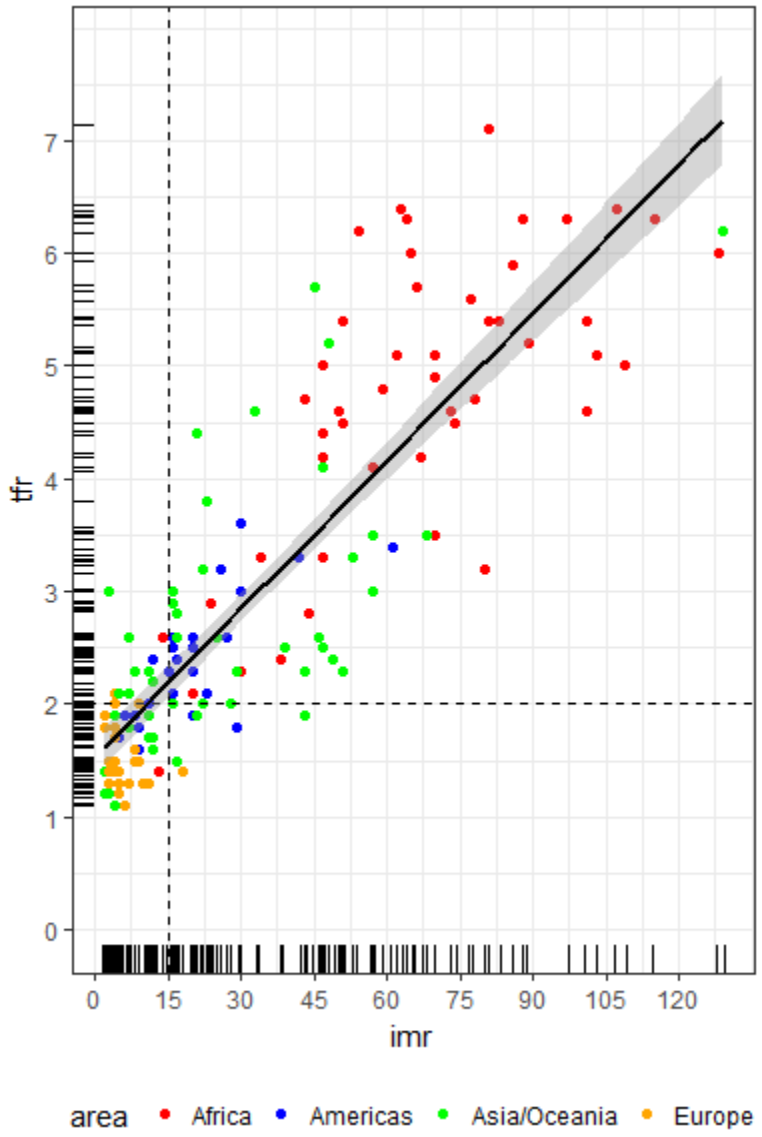



```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")
w$imrGT15 <- w$imr > 15
p <- ggplot(data=w,
            aes(x=area, fill=imrGT15))
p + geom_bar() +
scale_fill_manual(name="IMR value",
values = c("red","blue"),
labels=c("<=15", ">15")) +
  theme(legend.text.align=1)
```



Data + Statistical Summary + Annotation

```
w <- read.csv(file="WDS2012.csv",
              head=TRUE, sep=",")

p <- ggplot(data=w, aes(x=imr,y=tfr))
p + geom_point(aes(color=area)) +
scale_color_manual(values=
      c("red", "blue", "green", "orange")) +
scale_y_continuous(breaks=c(0,1,2,3,4,5,6,7),
                   limits=c(0,7.8)) +
scale_x_continuous(breaks=
        c(0,15,30,45,60,75,90,105,120)) +
theme_bw() +
theme(legend.position="bottom",
      legend.direction="horizontal",
      legend.key=element_blank()) +
geom_vline(xintercept=15,linetype="dashed") +
geom_hline(yintercept=2,linetype="dashed") +
geom_smooth(method="lm", color="black", size=.8)
geom_rug(position="jitter", size=.1)
```

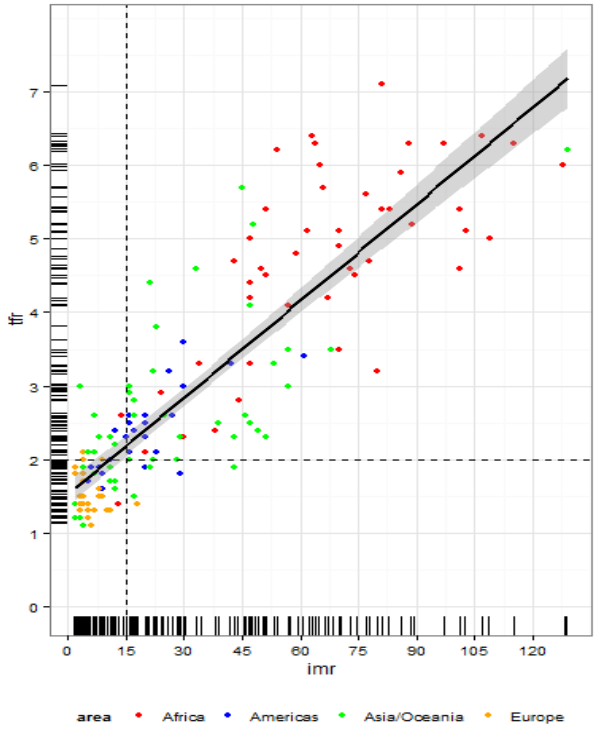


Part 3: Recap and Additional Resources

Recap

country	tfr	imr	area
Algeria	2.9	24	Africa
Egypt	2.9	24	Africa
.	.	.	.
.	.	.	.
.	.	.	.
Canada	1.7	5.1	Americas
United States	1.9	6.0	Americas
.	.	.	.
.	.	.	.
.	.	.	.
Armenia	1.7	11	Asia/Oceania
Azerbaijan	2.3	11	Asia/Oceania
.	.	.	.
.	.	.	.
.	.	.	.
Denmark	1.8	3.5	Europe
Estonia	2.5	3.3	Europe
.	.	.	.
.	.	.	.
.	.	.	.

ggplot2



construct graphs by considering:

- coordinate system
- which values will be represented by various visual characteristics (aesthetics)
- how values will mapped to visual characteristics (scales)
- geometric rendering (geom)
- whether data might be displayed as “small multiples” (facets)
- adding additional annotation

Additional ggplot2 Resources

official "Package ggplot2" documentation and help

- http://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
- http://ggplot2.tidyverse.org

online ggplot2 user community

- http://groups.google.com/group/ggplot2
- http://stackoverflow.com/tags/ggplot2
- https://rstudio.com/resources/cheatsheets/

books

- *ggplot2: Elegant Graphics for Data Analysis, Second Edition,* by Hadley Wickham. Springer, 2016.

- *R for Data Science (*Chapter 3*)*, by Hadley Wickham & Garrett Grolemund, online at https://r4ds.had.co.nz/data-visualisation.html , 2017.

- *Data Visualization : A practical introduction with R and ggplot2*, by Kieran Healy, online at http://socviz.co/, 2018.

- *R Graphics Cookbook, 2nd Edition* by Winston Chang, online at https://r-graphics.org/ , 2021.

- *The Grammar of Graphics* by Leland Wilkinson. Springer, 2005.

Thank You!

Workshop Survey

Computing Training: https://researchcomputing.princeton.edu/workshops

Help Sessions: Tuesdays 10:30-11:30am and Thursdays 2:00-3:00pm
https://researchcomputing.princeton.edu/education/help-sessions

Instructor email: