

BMS

R & bioinformatics tools

Installation of R



[\[Home\]](#)

Download

[CRAN](#)

R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting Bugs](#)

[Conferences](#)

[Search](#)

[Get Involved: Mailing Lists](#)

[Developer Pages](#)

[R Blog](#)

R Foundation

[Foundation](#)

[Board](#)

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- [R version 3.5.2 \(Eggshell Igloo\) prerelease versions](#) will appear starting Monday 2018-12-10. Final release is scheduled for Thursday 2018-12-20.
- The R Foundation Conference Committee has released a [call for proposals](#) to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a [supporting member](#)
- [R version 3.5.1 \(Feather Spray\)](#) has been released on 2018-07-02.
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

News via Twitter

<https://www.r-project.org/>

Installation of R

Korea	https://ftp.harukasan.org/CRAN/ https://cran.yu.ac.kr/ https://cran.seoul.go.kr/ http://healthstat.snu.ac.kr/CRAN/ https://cran.biodisk.org/ http://cran.biodisk.org/	Information and Database Systems Laboratory, Pukyong National University Yeungnam University Bigdata Campus, Seoul Metropolitan Government Graduate School of Public Health, Seoul National University, Seoul The Genome Institute of UNIST (Ulsan National Institute of Science and Technology) The Genome Institute of UNIST (Ulsan National Institute of Science and Technology)
Malaysia	https://wbc.upm.edu.my/cran/ http://wbc.upm.edu.my/cran/	Universiti Putra Malaysia Universiti Putra Malaysia
Mexico	https://cran.itam.mx/ http://cran.itam.mx/ http://www.est.colpos.mx/R-mirror/	Instituto Tecnológico Autónomo de México Instituto Tecnológico Autónomo de México Colegio de Postgraduados, Texcoco

Installation of R

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-07-05, Action of the Toes) [R-3.6.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Installation of R

R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to install R for the first time .
contrib	Binaries of contributed CRAN packages (for R $\geq$ 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old_contrib	Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.13.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Installation of R

[Download R 3.6.1 for Windows](#) (81 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

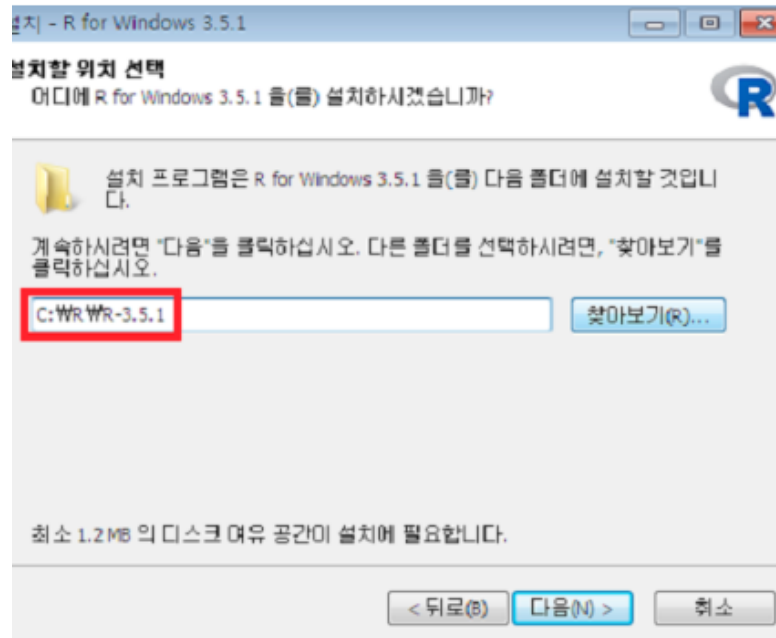
Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

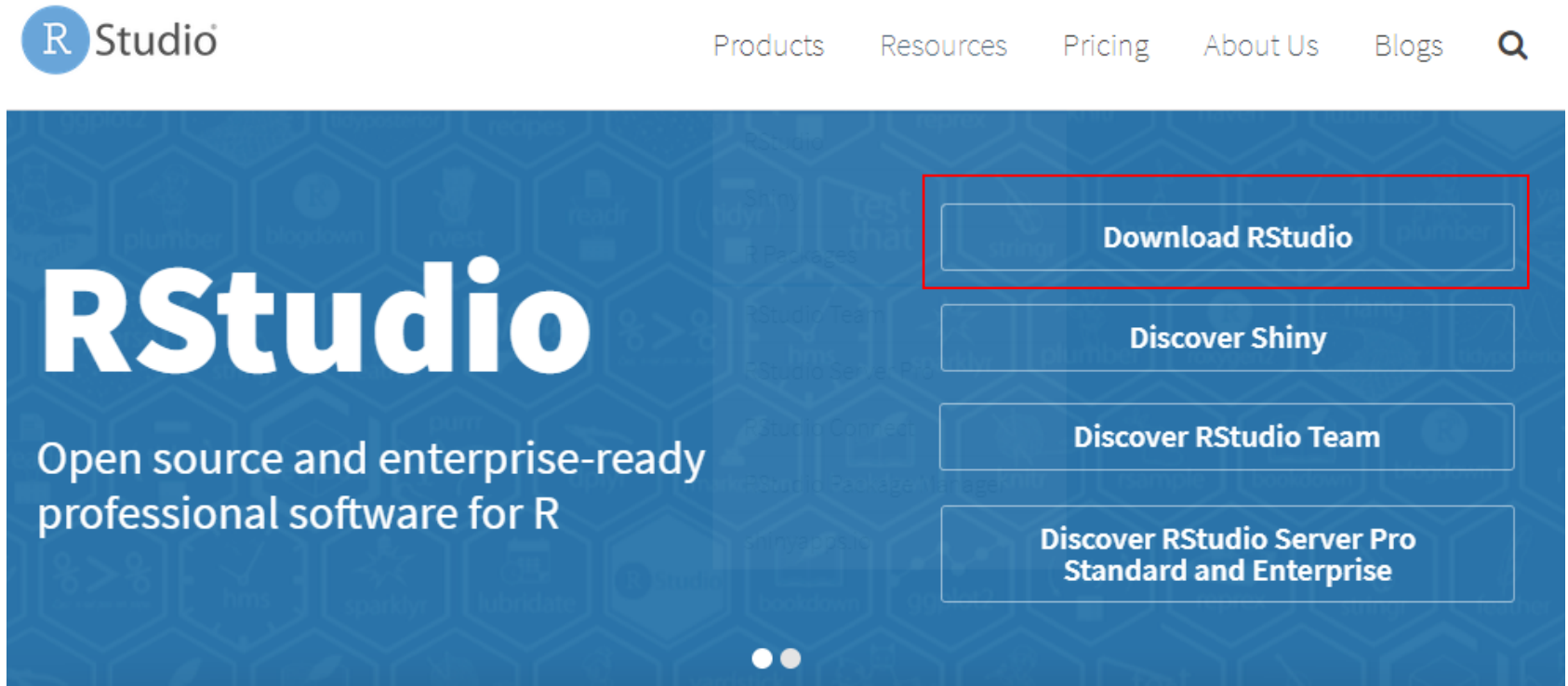
- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is CRAN.MIRROR>/bin/windows/base/release.htm.

Installation of R



Installation of RStudio



<https://www.rstudio.com/>

Installation of RStudio

 [Products](#) [Resources](#) [Pricing](#) [About Us](#) [Blogs](#) 

	RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Server Pro Commercial License
	FREE	\$995 per year	FREE	\$4,975 per year (5 Named Users)
	DOWNLOAD	BUY	DOWNLOAD	BUY
	Learn More	Learn More	Learn More	Evaluation Learn More
Integrated Tools for R	●	●	●	●
Priority Support		●		●
Access via Web Browser			●	●
Enterprise Security				●
Project Sharing				●
Manage Multiple R Sessions & Versions				●
Admin Dashboard				●
Load Balancing				●
Auditing and Monitoring				●
Data Connectivity				●
Launcher				●

Installation of RStudio

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - Mac OS X 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045ff13c7
RStudio 1.2.1335 - Ubuntu 18/Debian 10 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46c fb0aea75
RStudio 1.2.1335 - Fedora 19/RedHat 7 (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9 (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15 (64-bit)	101.6 MB	2019-04-08	2795a63c7efd8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12 (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eefcae1

Definition of Bioinformatics

Data Visualization with ggplot2 :: CHEAT SHEET

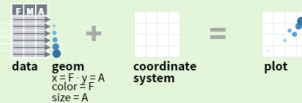


Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot (data = <DATA>) +  
  <GEOM_FUNCTION> (mapping = aes(<MAPPINGS>),  
    stat = <STAT>, position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

Not required, sensible defaults supplied

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings data geom
qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5" x 5" file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

- a + **geom_blank()** (Useful for expanding limits)
- b + **geom_curve**(aes(yend = lat + 1, xend = long + 1, curvature = z)) - x, yend, y, alpha, color, curvature, linetype, size
- a + **geom_path**(lineend = "butt", linejoin = "round", linemitre = 1) x, y, alpha, color, group, linetype, size
- a + **geom_polygon**(aes(group = group)) x, y, alpha, color, fill, group, linetype, size
- b + **geom_rect**(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size
- a + **geom_ribbon**(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

- common aesthetics: x, y, alpha, color, linetype, size
- b + **geom_abline**(aes(intercept = 0, slope = 1))
- b + **geom_hline**(aes(yintercept = lat))
- b + **geom_vline**(aes(xintercept = long))
- b + **geom_segment**(aes(yend = lat + 1, xend = long + 1))
- b + **geom_spoke**(aes(angle = 1:155, radius = 1))

ONE VARIABLE continuous

- c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
- c + **geom_area**(stat = "bin") x, y, alpha, color, fill, linetype, size
- c + **geom_density**(kernel = "gaussian") x, y, alpha, color, fill, group, linetype, size, weight
- c + **geom_dotplot**() x, y, alpha, color, fill
- c + **geom_freqpoly**() x, y, alpha, color, group, linetype, size
- c + **geom_histogram**(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight
- c2 + **geom_qq**(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete

- d <- ggplot(mpg, aes(f))
- d + **geom_bar**() x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

continuous x, continuous y
e <- ggplot(mpg, aes(cty, hwy))

- e + **geom_label**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
- e + **geom_jitter**(height = 2, width = 2) x, y, alpha, color, fill, shape, size
- e + **geom_point**() x, y, alpha, color, fill, shape, size, stroke
- e + **geom_quantile**() x, y, alpha, color, group, linetype, size, weight
- e + **geom_rug**(sides = "bl") x, y, alpha, color, color, fill, group, linetype, size, weight
- e + **geom_smooth**(method = lm) x, y, alpha, color, fill, group, linetype, size, weight
- e + **geom_text**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x, continuous y

- f <- ggplot(mpg, aes(class, hwy))
- f + **geom_col**() x, y, alpha, color, fill, group, linetype, size
- f + **geom_boxplot**() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight
- f + **geom_dotplot**(binaxis = "y", stackdir = "center") x, y, alpha, color, fill, group
- f + **geom_violin**(scale = "area") x, y, alpha, color, fill, group, linetype, size, weight

discrete x, discrete y

- g <- ggplot(diamonds, aes(cut, color))
- g + **geom_count**() x, y, alpha, color, fill, shape, size, stroke

THREE VARIABLES

- seals\$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))
- l + **geom_contour**(aes(z = z)) x, y, z, alpha, color, group, linetype, size, weight
- l + **geom_raster**(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE) x, y, alpha, fill
- l + **geom_tile**(aes(fill = z), x, y, alpha, color, fill, linetype, size, width)

continuous bivariate distribution

- h <- ggplot(diamonds, aes(carat, price))
- h + **geom_bin2d**(binwidth = c(0.25, 500)) x, y, alpha, color, fill, linetype, size, weight
- h + **geom_density2d**() x, y, alpha, color, group, linetype, size
- h + **geom_hex**() x, y, alpha, color, fill, size

continuous function

- i <- ggplot(economics, aes(date, unemploy))
- i + **geom_area**() x, y, alpha, color, fill, linetype, size
- i + **geom_line**() x, y, alpha, color, group, linetype, size
- i + **geom_step**(direction = "hv") x, y, alpha, color, group, linetype, size

visualizing error

- df <- data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2)
- j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))
- j + **geom_crossbar**(fatten = 2) x, y, ymax, ymin, alpha, color, fill, group, linetype, size
- j + **geom_errorbar**() x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh**())
- j + **geom_linerange**() x, ymin, ymax, alpha, color, group, linetype, size
- j + **geom_pointrange**() x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

- data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests)))
- map <- map_data("state")
- k <- ggplot(data, aes(fill = murder))
- k + **geom_map**(aes(map_id = state), map = map) + **expand_limits**(x = map\$long, y = map\$lat), map_id, alpha, color, fill, linetype, size



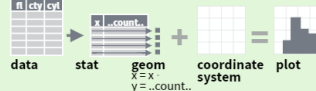
RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at <http://ggplot2.tidyverse.org> • ggplot2 3.1.0 • Updated: 2018-12

Definition of Bioinformatics

Stats

An alternative way to build a layer

A stat builds new variables to plot (e.g., count, prop).



Visualize a stat by changing the default stat of a geom function, `geom_bar(stat="count")` or by using a stat function, `stat_count(geom="bar")`, which calls a default geom to make a layer (equivalent to a geom function). Use `..name..` syntax to map stat variables to aesthetics.



```
c + stat_bin(binwidth = 1, origin = 10)
x, y | ..count.., ..ncount.., ..density.., ..ndensity..
c + stat_count(width = 1) x, y | ..count.., ..prop..
c + stat_density(adjust = 1, kernel = "gaussian")
x, y | ..count.., ..density.., ..scaled..
```

```
e + stat_bin_2d(bins = 30, drop = T)
x, y, fill | ..count.., ..density..
```

```
e + stat_bin_hex(bins = 30) x, y, fill | ..count.., ..density..
```

```
e + stat_density_2d(contour = TRUE, n = 100)
x, y, color, size | ..level..
```

```
e + stat_ellipse(level = 0.95, segments = 51, type = "t")
```

```
l + stat_contour(aes(z = z)) x, y, z, order | ..level..
```

```
l + stat_summary_hex(aes(z = z), bins = 30, fun = max)
x, y, z, fill | ..value..
```

```
l + stat_summary_2d(aes(z = z), bins = 30, fun = mean)
x, y, z, fill | ..value..
```

```
f + stat_boxplot(coef = 1.5) x, y | ..lower.., ..middle.., ..upper.., ..width.., ..ymin.., ..ymax..
```

```
f + stat_ydensity(kernel = "gaussian", scale = "area") x, y | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..
```

```
e + stat_ecdf(n = 40) x, y | ..x.., ..y..
```

```
e + stat_quantile(quantiles = c(0.1, 0.9), formula = y ~ x, method = "rq") x, y | ..quantile..
```

```
e + stat_smooth(method = "lm", formula = y ~ x, se = T, level = 0.95) x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..
```

```
ggplot() + stat_function(aes(x = -3:3), n = 99, fun = dnorm, args = list(sd = 0.5)) x | ..x.., ..y..
```

```
e + stat_identity(na.rm = TRUE)
```

```
ggplot() + stat_qq(aes(sample = 1:100), dist = qt, dparam = list(df = 5)) sample, x, y | ..sample.., ..theoretical..
```

```
e + stat_sum() x, y, size | ..n.., ..prop..
```

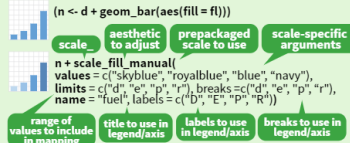
```
e + stat_summary(fun.data = "mean_cl_boot")
```

```
h + stat_summary_bin(fun.y = "mean", geom = "bar")
```

```
e + stat_unique()
```

Scales

Scales map data values to the visual values of an aesthetic. To change a mapping, add a new scale.



GENERAL PURPOSE SCALES

Use with most aesthetics

`scale_*` `continuous()` - map cont' values to visual ones

`scale_*` `discrete()` - map discrete values to visual ones

`scale_*` `identity()` - use data values as visual ones

`scale_*` `manual(values = c())` - map discrete values to manually chosen visual ones

`scale_*` `date(date_labels = "%m/%d")`, `date_breaks = "2 weeks"` - treat data values as dates.

`scale_*` `datetime()` - treat data x values as data times.

Use same arguments as `scale_x_date()`. See ?strptime for label formats.

X & Y LOCATION SCALES

Use with x or y aesthetics (x shown here)

`scale_x_log10()` - Plot x on log10 scale

`scale_x_reverse()` - Reverse direction of x axis

`scale_x_sqrt()` - Plot x on square root scale

COLOR AND FILL SCALES (DISCRETE)

`n <- d + geom_bar(aes(fill = fl))`

`n + scale_fill_brewer(palette = "Blues")`

For palette choices: `RColorBrewer::display.brewer.all()`

`n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red")`

COLOR AND FILL SCALES (CONTINUOUS)

`o <- c + geom_dotplot(aes(fill = ..x..))`

`o + scale_fill_distiller(palette = "Blues")`

`o + scale_fill_gradient(low = "red", high = "yellow")`

`o + scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 25)`

`o + scale_fill_gradientn(colours = topo.colors(6))`

Also: `rainbow()`, `heat.colors()`, `terrain.colors()`, `cm.colors()`, `RColorBrewer::brewer.pal()`

SHAPE AND SIZE SCALES

`p <- e + geom_point(aes(shape = fl, size = cyl))`

`p + scale_shape() + scale_size()`

`p + scale_shape_manual(values = c(3:7))`

`p + scale_size(range = c(1,6))`

`p + scale_size_area(max_size = 6)`

Coordinate Systems

`r <- d + geom_bar()`

`r + coord_cartesian(xlim = c(0, 5))`

`xlim, ylim`

The default cartesian coordinate system

`r + coord_fixed(ratio = 1/2)`

`ratio, xlim, ylim`

Cartesian coordinates with fixed aspect ratio between x and y units

`r + coord_flip()`

`xlim, ylim`

Flipped Cartesian coordinates

`r + coord_polar(theta = "x", direction = 1)`

`theta, start, direction`

Polar coordinates

`r + coord_trans(ytrans = "sqrt")`

`xtrans, ytrans, xlim, ylim`

Transformed cartesian coordinates. Set xtrans and ytrans to the name of a window function.

`π + coord_quickmap()`

`π + coord_map(projection = "ortho", orientation = c(41, -74, 0))`

`xlim, ylim`

Map projections from the mapproj package (mercator (default), azequalarea, lagrange, etc.)

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

`s <- ggplot(mpg, aes(fl, fill = drv))`

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

Arrange elements side by side

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

Stack elements on top of one another, normalize height

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

Add random noise to X and Y position of each element to avoid overplotting

`e + geom_label(position = "nudge")`

Nudge labels away from points

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

Stack elements on top of one another

Each position adjustment can be recast as a function with manual width and height arguments

`s + geom_bar(position = position_dodge(width = 1))`

Themes

`r + theme_bw()`

White background with grid lines

`r + theme_classic()`

`r + theme_light()`

`r + theme_minimal()`

Minimal themes

`r + theme_void()`

Empty theme

Faceting

Facets divide a plot into subplots based on the values of one or more discrete variables.

`t <- ggplot(mpg, aes(cty, hwy)) + geom_point()`

`t + facet_grid(cols = vars(fl))`

facet into columns based on fl

`t + facet_grid(rows = vars(year))`

facet into rows based on year

`t + facet_grid(rows = vars(year), cols = vars(fl))`

facet into both rows and columns

`t + facet_wrap(vars(fl))`

wrap facets into a rectangular layout

Set scales to let axis limits vary across facets

`t + facet_grid(rows = vars(drv), cols = vars(fl), scales = "free")`

x and y axis limits adjust to individual facets

`"free_x"` - x axis limits adjust

`"free_y"` - y axis limits adjust

Set labeller to adjust facet labels

`t + facet_grid(cols = vars(fl), labeller = label_both)`

`fl:c fl:d fl:e fl:f fl:r`

`t + facet_grid(rows = vars(fl), labeller = label_bquote(alpha ^ .(fl)))`

`αc αd αe αf αr`

Labels

`t + labs(x = "New x axis label", y = "New y axis label", title = "Add a title above the plot", subtitle = "Add a subtitle below title", caption = "Add a caption below plot", <AES> = "New <AES> legend title")`

Use scale functions to update legend labels

`t + annotate(geom = "text", x = 8, y = 9, label = "A")`

geom to place manual values for geom's aesthetics

Legends

`n + theme(legend.position = "bottom")`

Place legend at "bottom", "top", "left", or "right"

`n + guides(fill = "none")`

Set legend type for each aesthetic: colorbar, legend, or none (no legend)

`n + scale_fill_discrete(name = "Title", labels = c("A", "B", "C", "D", "E"))`

Set legend title and labels with a scale function.

Zooming

Without clipping (preferred)

`t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20))`

With clipping (removes unseen data points)

`t + xlim(0, 100) + ylim(10, 20)`

`t + scale_x_continuous(limits = c(0, 100)) + scale_y_continuous(limits = c(10, 20))`



Grammar of Graphics

ggplot2 is an R package for producing statistical, or data, graphics, but it is unlike most other graphics packages because it has a deep underlying grammar. This grammar, based on the Grammar of Graphics (Wilkinson 2005), is made up of a set of independent components that can be composed in many different ways. This makes ggplot2 very powerful because you are not limited to a set of pre-specified graphics, but you can create new graphics that are precisely tailored for your problem. This may sound overwhelming, but because there is a simple set of core principles and very few special cases, ggplot2 is also easy to learn (although it may take a little time to forget your preconceptions from other graphics tools).

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

Load package and mpg data

```
library(ggplot2)
```

```
head(mpg)
```

```
# A tibble: 6 x 11
  manufacturer model displ  year   cyl trans      drv   cty   hwy fl   class
  <chr>         <chr> <dbl> <int> <int> <chr>    <chr> <int> <int> <chr> <chr>
1 audi         a4      1.8  1999     4 auto(l5) f       18    29 p    compact
2 audi         a4      1.8  1999     4 manual(m5) f       21    29 p    compact
3 audi         a4      2    2008     4 manual(m6) f       20    31 p    compact
4 audi         a4      2    2008     4 auto(av) f       21    30 p    compact
5 audi         a4      2.8  1999     6 auto(l5) f       16    26 p    compact
6 audi         a4      2.8  1999     6 manual(m5) f       18    26 p    compact
```

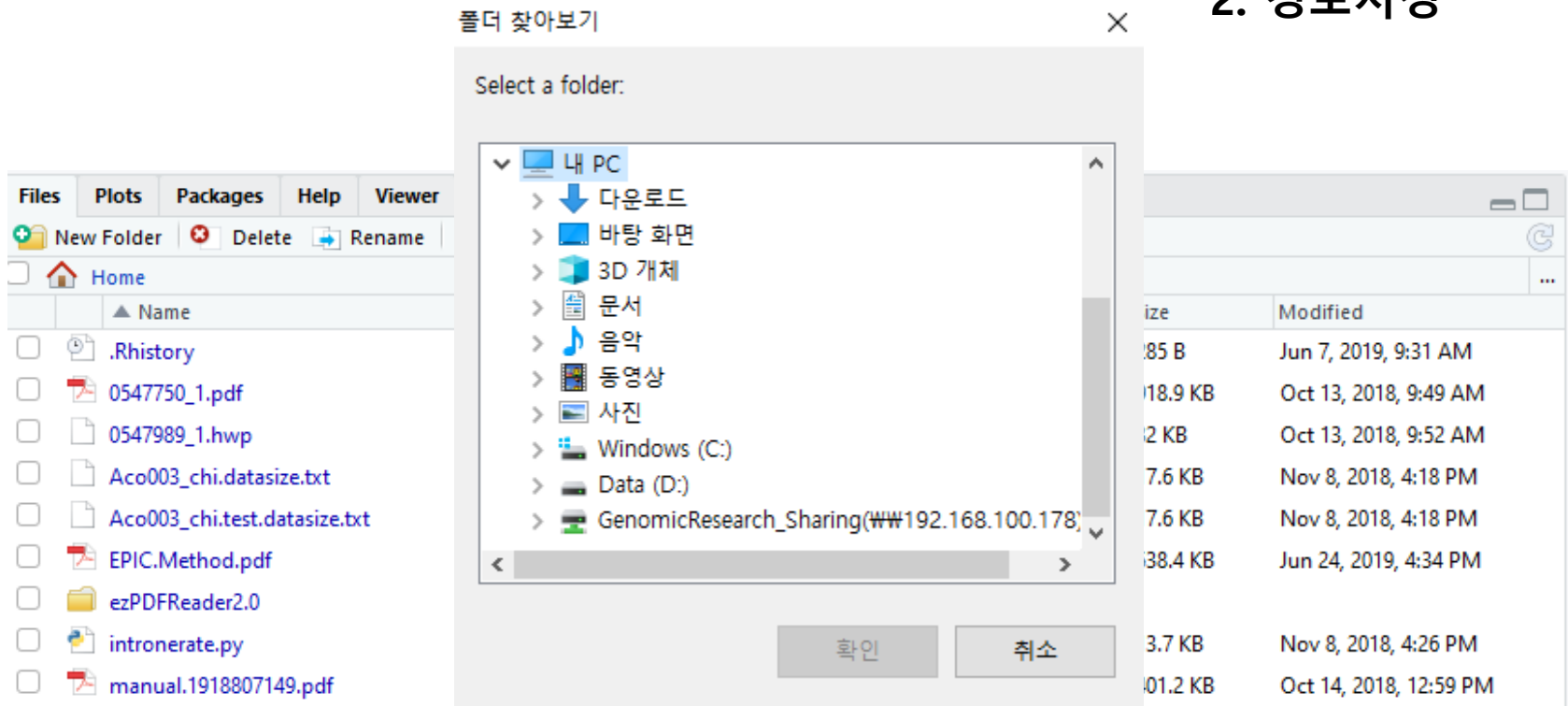
Change path

1.클릭



Change path

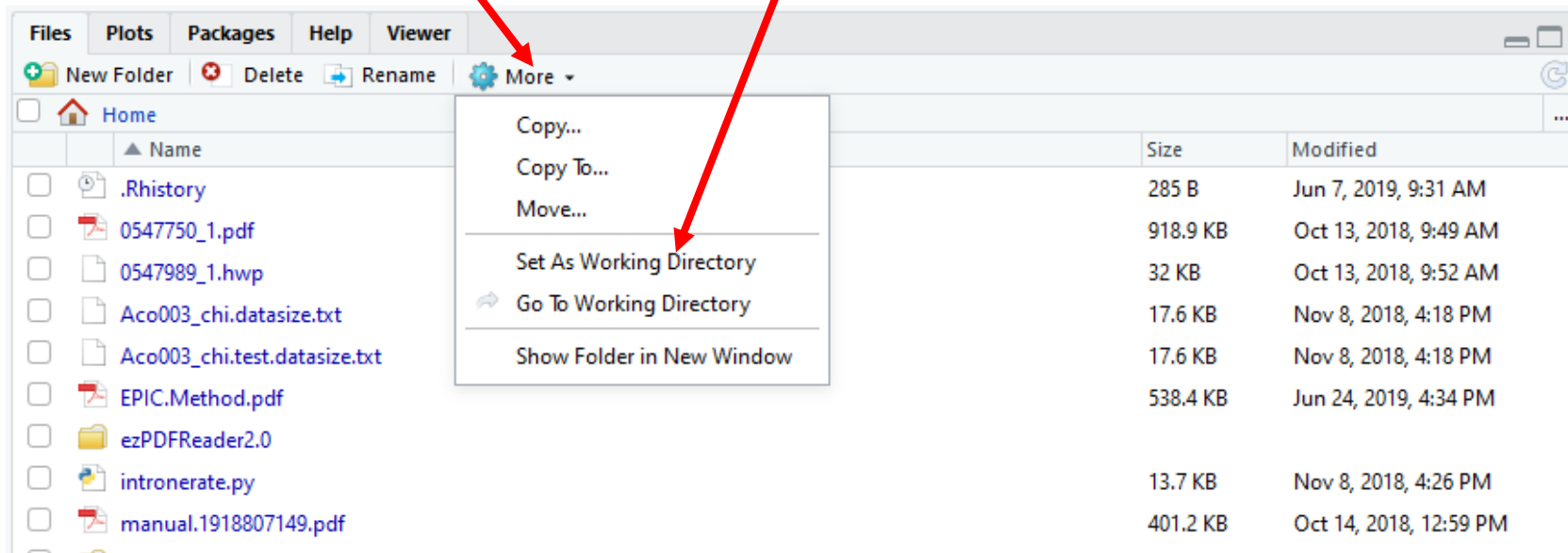
2. 경로지정



Change path

3.클릭

4.클릭(경로 설정)



mpg data

```
write.table(mpg,"mpg.txt",sep="Wt")
```

manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
audi	a4 quattro	2	2008	4	manual(m6)	4	20	28	p	compact
audi	a4 quattro	2	2008	4	auto(s6)	4	19	27	p	compact
audi	a4 quattro	2.8	1999	6	auto(l5)	4	15	25	p	compact
audi	a4 quattro	2.8	1999	6	manual(m5)	4	17	25	p	compact
audi	a4 quattro	3.1	2008	6	auto(s6)	4	17	25	p	compact

Data information

summary(mpg)

```
manufacturer      model      displ      year      cyl      trans
Length:234      Length:234      Min.   :1.600      Min.   :1999      Min.   :4.000      Length:234
Class :character  Class :character  1st Qu.:2.400      1st Qu.:1999      1st Qu.:4.000      Class :character
Mode  :character  Mode  :character  Median :3.300      Median :2004      Median :6.000      Mode  :character
                  Mean   :3.472      Mean   :2004      Mean   :5.889
                  3rd Qu.:4.600      3rd Qu.:2008      3rd Qu.:8.000
                  Max.   :7.000      Max.   :2008      Max.   :8.000

      drv      cty      hwy      fl      class
Length:234      Min.   : 9.00      Min.   :12.00      Length:234      Length:234
Class :character  1st Qu.:14.00      1st Qu.:18.00      Class :character  Class :character
Mode  :character  Median :17.00      Median :24.00      Mode  :character  Mode  :character
                  Mean   :16.86      Mean   :23.44
                  3rd Qu.:19.00      3rd Qu.:27.00
                  Max.   :35.00      Max.   :44.00
```

model - 모델
displ - 배기량
year - 생산연도
cyl - 실린더 개수
trans - 변속기 종류

drv - 구동 방식
cty - 도시 연비
hwy - 고속도로 연비
fl - 연료 종류
class - 자동차 종류

Basic setting – ggplot2

```
ggplot(dataset, aes(x = x_data, y = y_data, color = color_group))
```

```
ggplot(mtcars, aes(x = wt))
```

```
ggplot(mtcars, aes(x = wt, y = mpg))
```

```
ggplot(mtcars, aes(x = wt, y = mpg))
```

- `aes(aesthetics)` = 미학요소
- `ggplot2`에서는 색상, 크기, 모양, 채우기 등과 함께 플롯의 X 축과 Y 축을 미학으로 간주

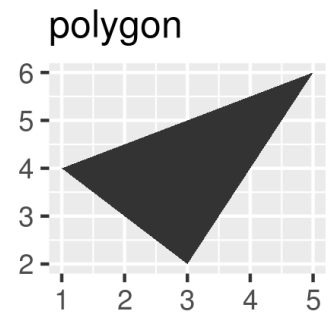
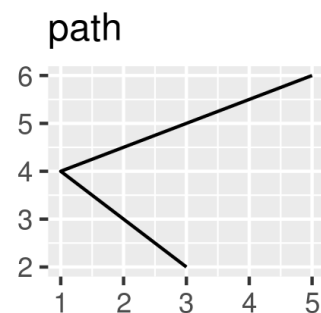
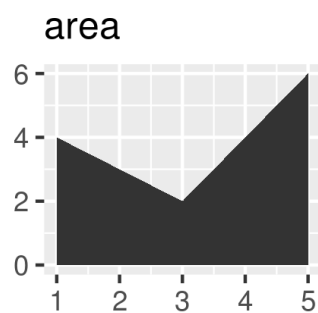
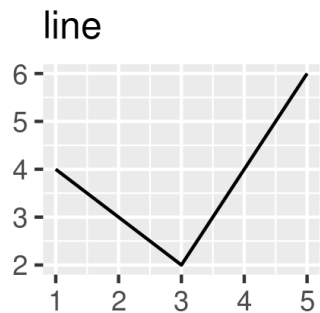
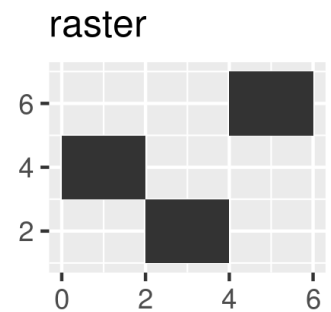
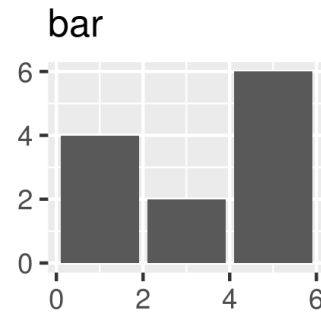
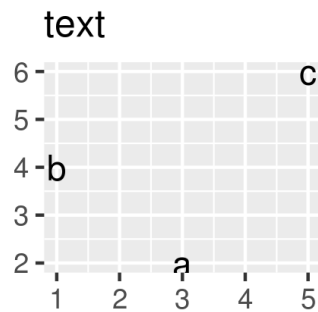
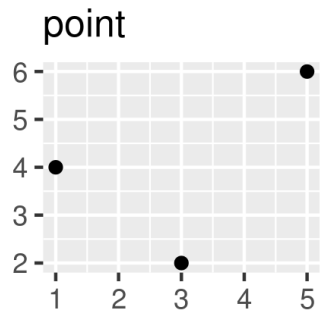
Layer – ggplot2

- Layer = '기하 구조(geoms)
- 기본 설정 후 기하 구조를 다른 구성 위에 추가 할 수 있으며, 이러한 추가는 '+' 기호를 사용함. 해당 링크에는 사용 가능한 모든 기하 구조의 목록이 있습니다. (Layers: geoms 참조)
- 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, ...

geom_xxx() – ggplot2

"geom_abline"	"Line specified by slope and intercept."
"geom_area"	"Area plot."
"geom_bar"	"Bars, rectangles with bases on x-axis"
"geom_bin2d"	"Add heatmap of 2d bin counts."
"geom_blank"	"Blank, draws nothing."
"geom_boxplot"	"Box and whiskers plot."
"geom_contour"	"Display contours of a 3d surface in 2d."
"geom_crossbar"	"Hollow bar with middle indicated by horizontal line."
"geom_density"	"Display a smooth density estimate."
"geom_density2d"	"Contours from a 2d density estimate."
"geom_dotplot"	"Dot plot"
"geom_errorbar"	"Error bars."
"geom_errorbarh"	"Horizontal error bars"
"geom_freqpoly"	"Frequency polygon."
"geom_hex"	"Hexagon binning."
"geom_histogram"	"Histogram"
"geom_hline"	"Horizontal line."
"geom_jitter"	"Points, jittered to reduce overplotting."
"geom_line"	"Connect observations, ordered by x value."
"geom_linerange"	"An interval represented by a vertical line."
"geom_map"	"Polygons from a reference map."
"geom_path"	"Connect observations in original order"
"geom_point"	"Points, as for a scatterplot"
"geom_pointrange"	"An interval represented by a vertical line, with a point in the middle."
"geom_polygon"	"Polygon, a filled path."
"geom_quantile"	"Add quantile lines from a quantile regression."
"geom_raster"	"High-performance rectangular tiling."
"geom_rect"	"2d rectangles."
"geom_ribbon"	"Ribbons, y range with continuous x values."
"geom_rug"	"Marginal rug plots."
"geom_segment"	"Single line segments."
"geom_smooth"	"Add a smoothed conditional mean."
"geom_step"	"Connect observations by stairs."
"geom_text"	"Textual annotations."
"geom_tile"	"Tile plane with rectangles."
"geom_violin"	"Violin plot."
"geom_vline"	"Line, vertical."

Individual geoms – ggplot2



Individual geoms – ggplot2

```
p <- ggplot(mpg, aes(displ, cty, label = model)) + labs(x = NULL, y =  
NULL) + theme(plot.title = element_text(size = 12))
```

```
p + geom_point() + ggtitle("point")
```

```
p + geom_text() + ggtitle("text")
```

```
p + geom_tile() + ggtitle("raster")
```

```
p + geom_tile() + ggtitle("raster")
```

```
p + geom_line() + ggtitle("line")
```

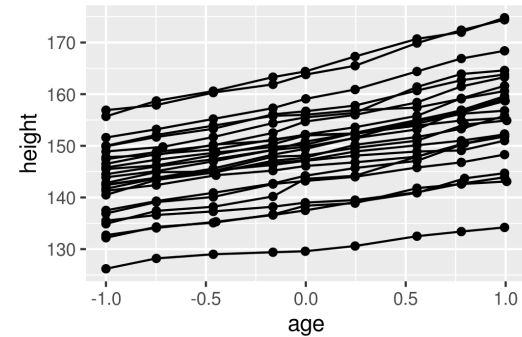
```
p + geom_area() + ggtitle("area")
```

```
p + geom_path() + ggtitle("path")
```

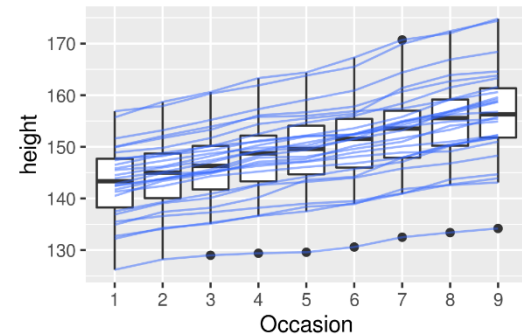
```
p + geom_polygon() + ggtitle("polygon")
```

Collective geoms – ggplot2

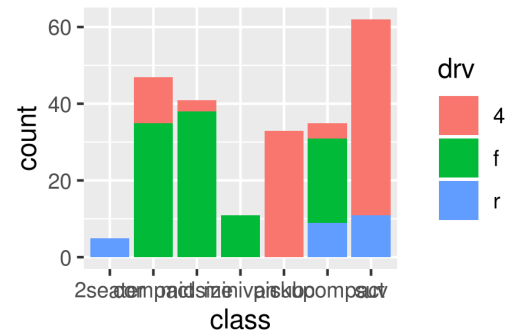
```
ggplot(data, aes(x, y)) + geom_point()  
+ geom_line()
```



```
ggplot(data, aes(x, y)) + geom_boxplot()  
+ geom_line(aes(group = group),  
colour = "#3366FF", alpha = 0.5)
```



```
ggplot(data, aes(x, fill = fill_group)) +  
geom_bar()
```



Collective geoms – ggplot2

```
ggplot(mpg, aes(cty, displ, group = manufacturer)) + geom_point() +  
geom_line()
```

```
ggplot(mpg, aes(class, cty))+ geom_boxplot() + geom_line( aes(group  
= cty), colour = "#3366FF", alpha = 0.5)
```

```
ggplot(mpg, aes(class, fill = drv)) + geom_bar()
```

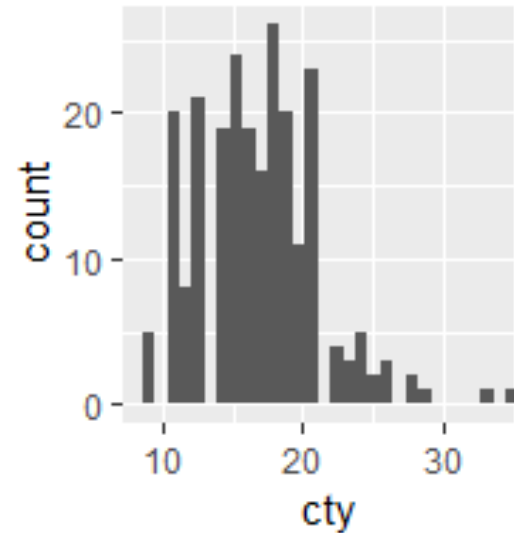
stat_xxx() – ggplot2

<code>"stat_abline"</code>	<code>"Add a line with slope and intercept."</code>
<code>"stat_bin"</code>	<code>"Bin data."</code>
<code>"stat_bin2d"</code>	<code>"Count number of observation in rectangular bins."</code>
<code>"stat_bindot"</code>	<code>"Bin data for dot plot."</code>
<code>"stat_binhex"</code>	<code>"Bin 2d plane into hexagons."</code>
<code>"stat_boxplot"</code>	<code>"Calculate components of box and whisker plot."</code>
<code>"stat_contour"</code>	<code>"Calculate contours of 3d data."</code>
<code>"stat_density"</code>	<code>"1d kernel density estimate."</code>
<code>"stat_density2d"</code>	<code>"2d density estimation."</code>
<code>"stat_ecdf"</code>	<code>"Empirical Cumulative Density Function"</code>
<code>"stat_function"</code>	<code>"Superimpose a function."</code>
<code>"stat_hline"</code>	<code>"Add a horizontal line"</code>
<code>"stat_identity"</code>	<code>"Identity statistic."</code>
<code>"stat_qq"</code>	<code>"Calculation for quantile-quantile plot."</code>
<code>"stat_quantile"</code>	<code>"Continuous quantiles."</code>
<code>"stat_smooth"</code>	<code>"Add a smoother."</code>
<code>"stat_spoke"</code>	<code>"Convert angle and radius to xend and yend."</code>
<code>"stat_sum"</code>	<code>"Sum unique values. Useful for overplotting on scatterplots."</code>
<code>"stat_summary"</code>	<code>"Summarise y values at every unique x."</code>
<code>"stat_summary2d"</code>	<code>"Apply function for 2D rectangular bins."</code>
<code>"stat_summary_hex"</code>	<code>"Apply function for 2D hexagonal bins."</code>
<code>"stat_unique"</code>	<code>"Remove duplicates."</code>
<code>"stat_vline"</code>	<code>"Add a vertical line"</code>
<code>"stat_ydensity"</code>	<code>"1d kernel density estimate along y axis, for violin plot."</code>

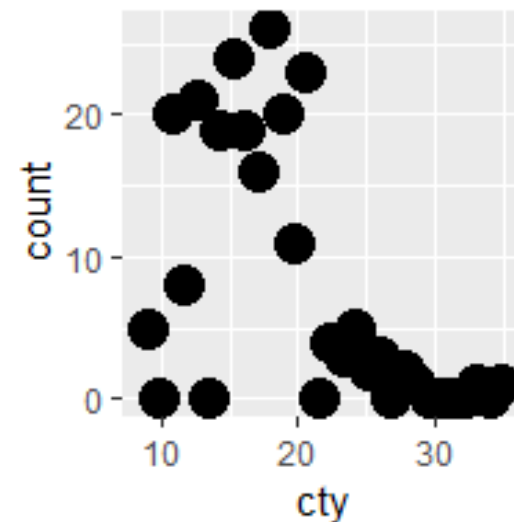
Statistical Transformation – ggplot2

```
p <- ggplot(mpg, aes(x=cty))
```

```
p + stat_bin()
```



```
p + stat_bin(geom="tile")
```



Change Default Geometric Object – ggplot2

```
p <- ggplot(mpg, aes(x=cty))
```

```
#figure1
```

```
p + stat_bin()
```

```
#figure2
```

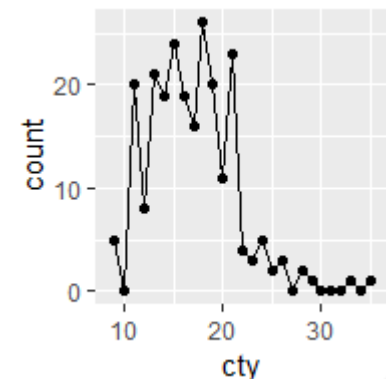
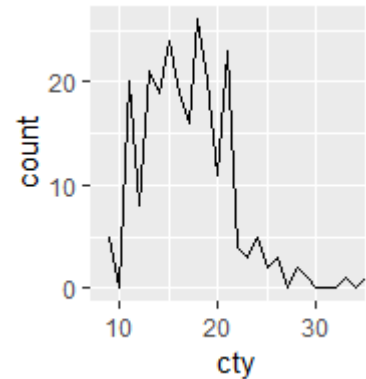
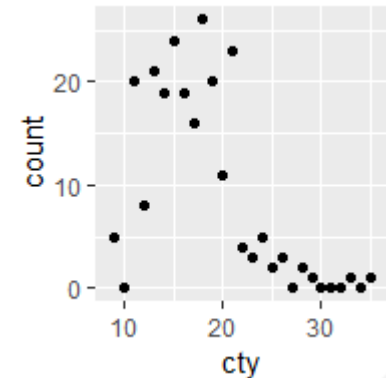
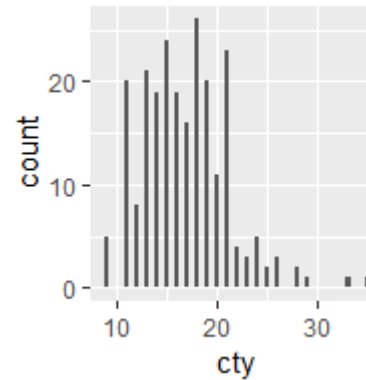
```
p + stat_bin(geom="point", binwidth=1)
```

```
#figure3
```

```
p + stat_bin(geom="line", binwidth=1)
```

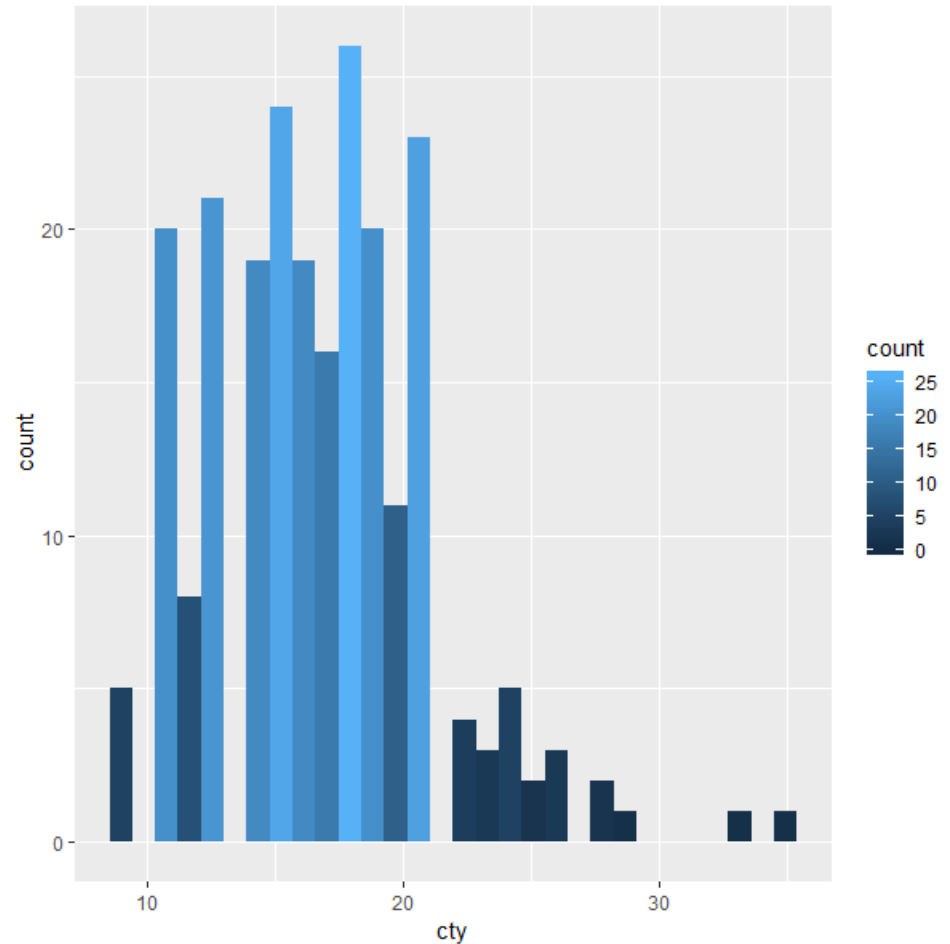
```
#figure4
```

```
p + stat_bin(geom="line",binwidth=1) +  
stat_bin(geom="point",binwidth=1)
```



Use Variables Created by stat_xxx() – ggplot2

```
p <- ggplot(mpg, aes(x=cty))  
p + stat_bin(aes(fill=..count..))
```



Add or Remove Aesthetic Mapping– ggplot2

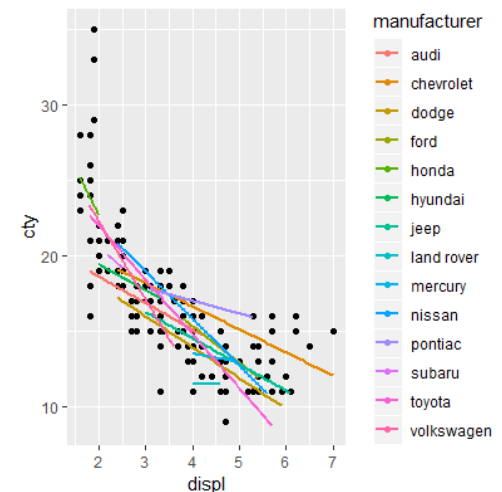
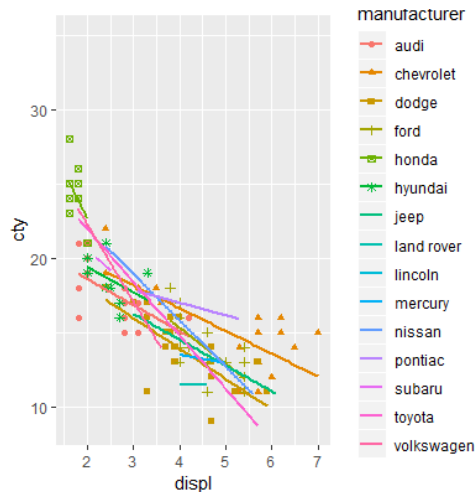
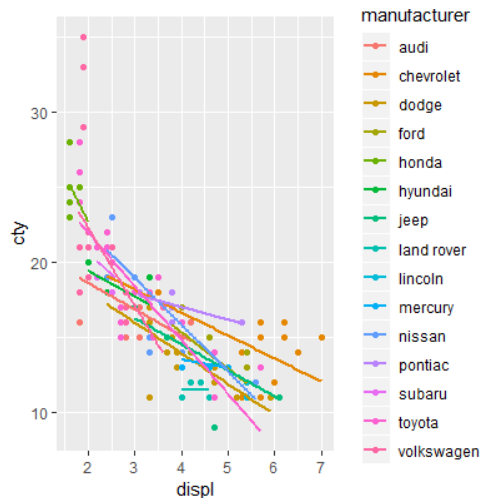
- Describe visual characteristics that represent data
 - for example, x position, y position, size, color (outside), fill (inside), point shape, line type, transparency
- Each layer inherits default aesthetics from plot object
 - within each layer, aesthetics may added, overwritten, or removed

```
p <- ggplot(mpg, aes(x=displ, y=cty, color=manufacturer))
```

```
p + geom_point() + geom_smooth(method="lm", se=FALSE) #fig1
```

```
p + geom_point(aes(shape=manufacturer)) + geom_smooth(method="lm",se=FALSE) #fig2
```

```
p + geom_point(aes(color=NULL)) + geom_smooth(method="lm",se=FALSE) #fig3
```



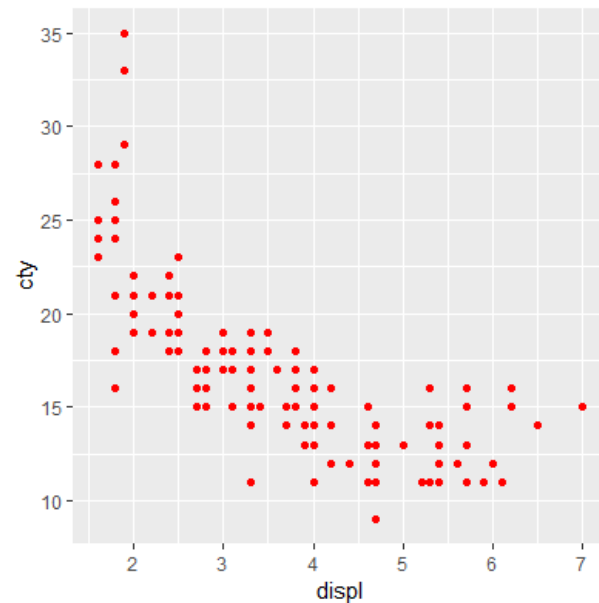
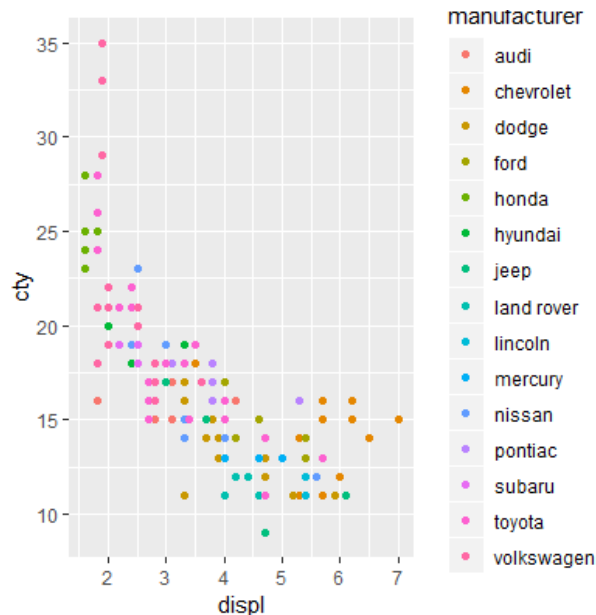
Aesthetic Mapping vs. Parameter Setting – ggplot2

- **aesthetic mapping**
 - data value determines visual characteristic
use `aes()`
- **Setting**
 - constant value determines visual characteristic
use layer parameter

```
p <- ggplot(mpg, aes(x=displ, y=cty))
```

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

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



Position – ggplot2

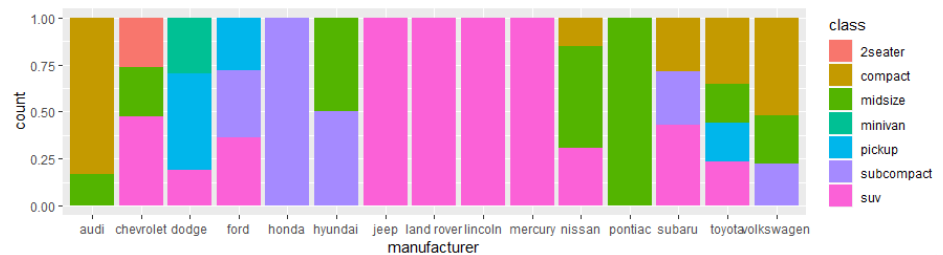
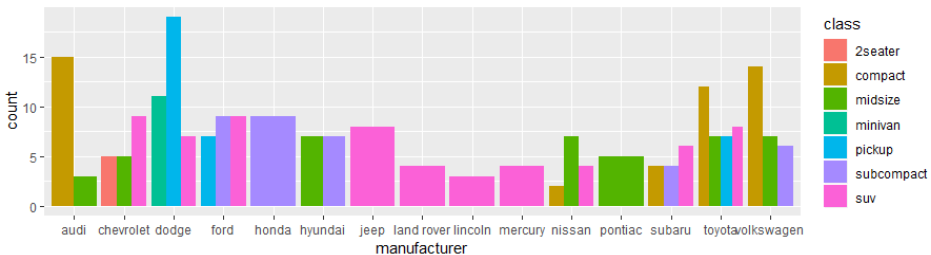
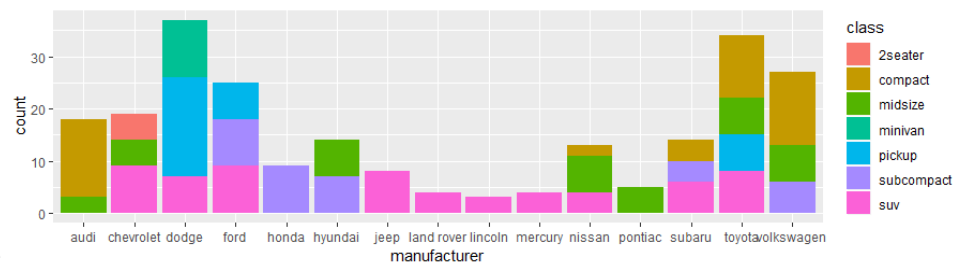
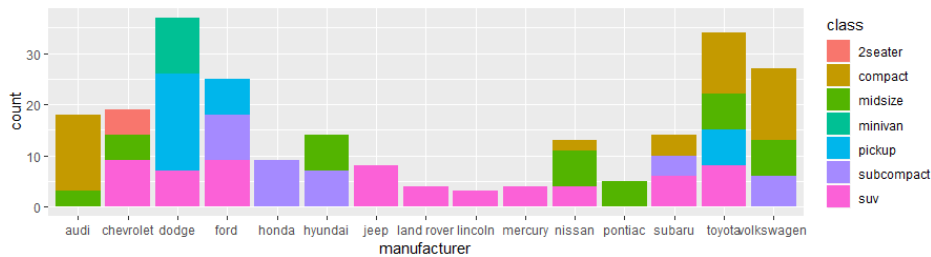
```
p <- ggplot(mpg, aes(x=manufacturer, fill=class))
```

```
p + geom_bar() #figure1
```

```
p + geom_bar(position="stack") #figure2
```

```
p + geom_bar(position="dodge") #figure3
```

```
p + geom_bar(position="fill") #figure4
```



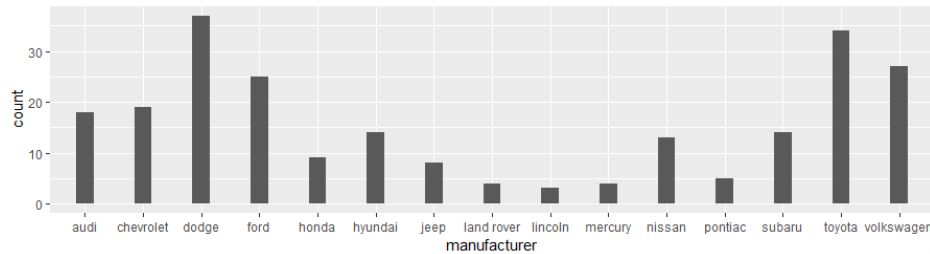
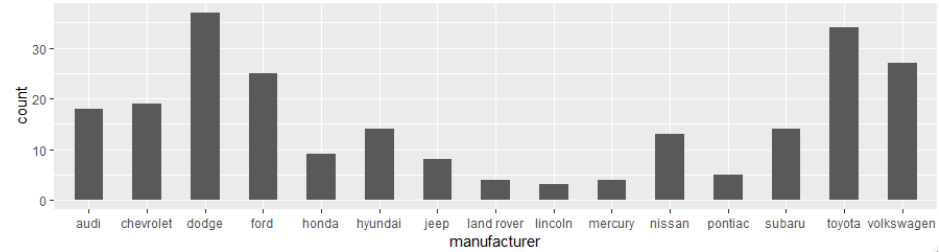
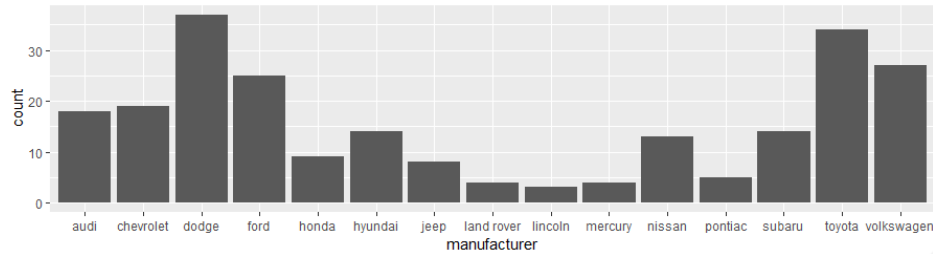
Bar Width – ggplot2

```
p <- ggplot(mpg, aes(x=manufacturer))
```

```
p + geom_bar()
```

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

```
p + geom_bar(width=.3)
```



Coordinate System – ggplot2

```
p <- ggplot(mpg, aes(x=factor(1),  
fill=manufacturer))
```

```
#figure1
```

```
p + geom_bar()
```

```
#figure2
```

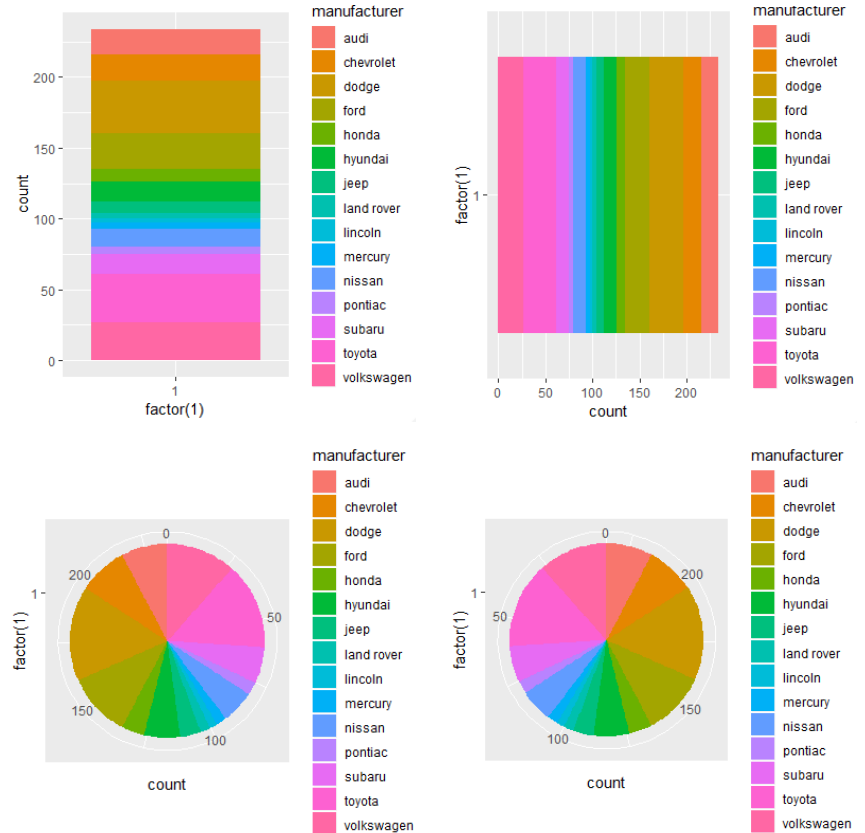
```
p + geom_bar() + coord_flip()
```

```
#figure3
```

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

```
#figure4
```

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



Fill Scales – ggplot2

```
p <- ggplot(mpg, aes(x=manufacturer, fill=class))
```

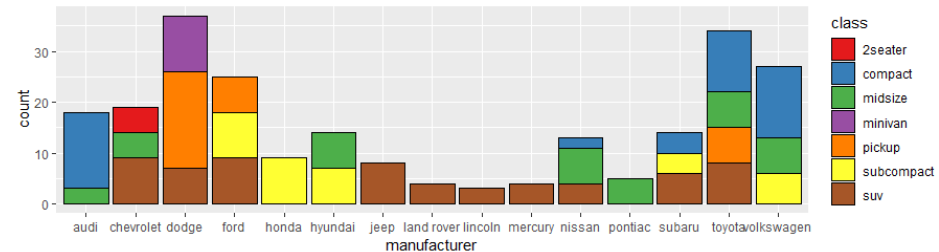
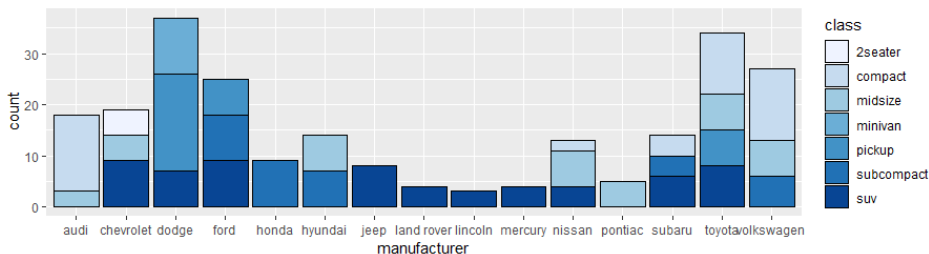
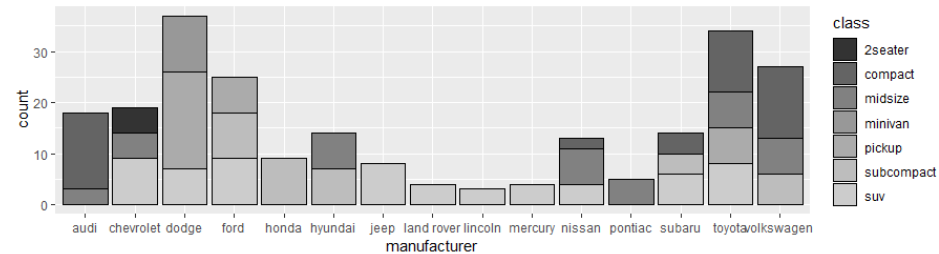
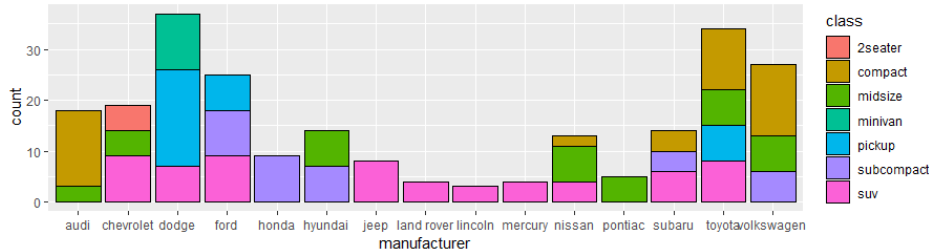
```
p + geom_bar(color="black") #figure1
```

```
p + geom_bar(color="black") + scale_fill_grey() #figure2
```

```
p + geom_bar(color="black") + scale_fill_brewer() #figure3
```

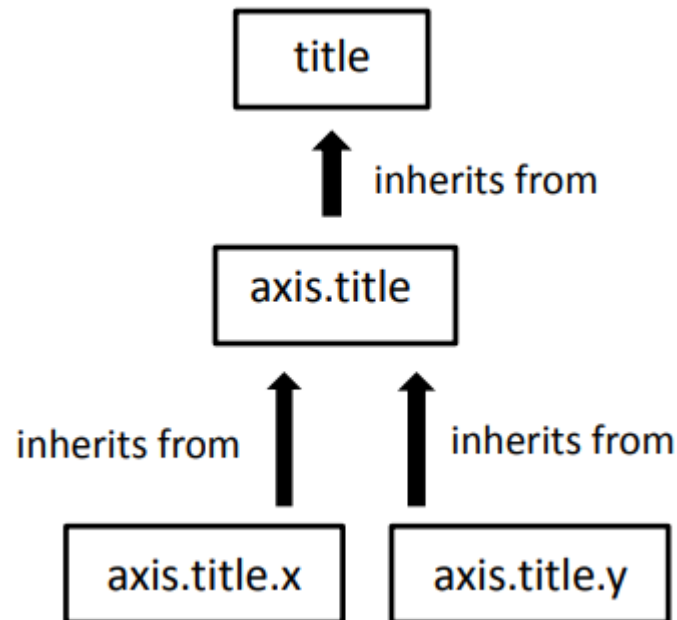
```
library(RColorBrewer)
```

```
p + geom_bar(color="black") + scale_fill_brewer(palette="Set1") #figure4
```



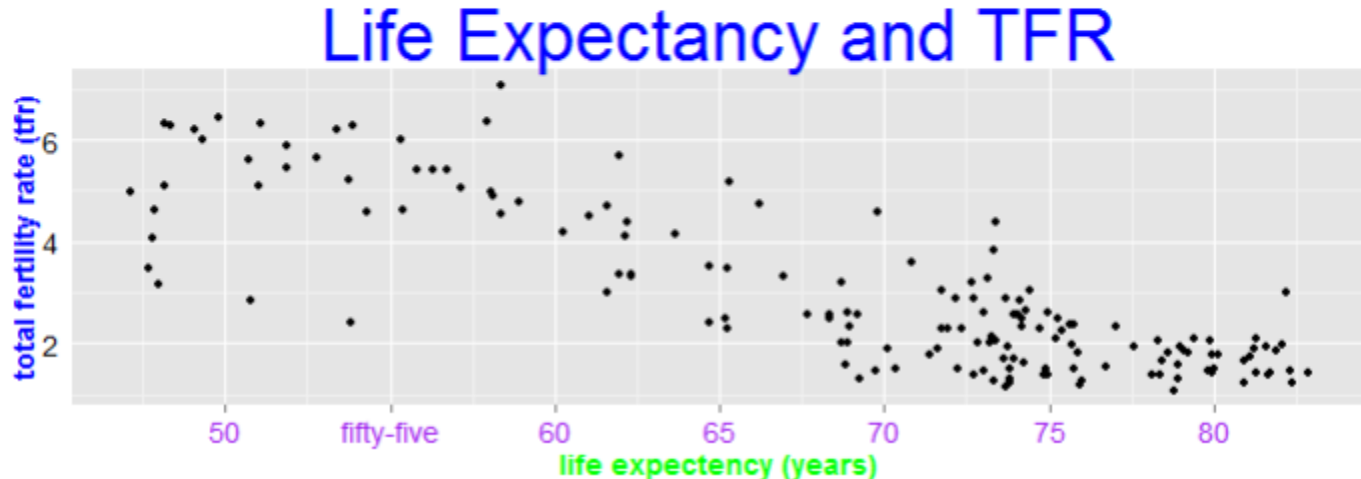
Theme – ggplot2

- Theme is non-data elements and
- Theme does not affect how data is displayed by `geom_xxx()` or `stat_xxx()` functions
- addition/modification/deletion of titles, axis labels, tick marks, axis tick labels and legends

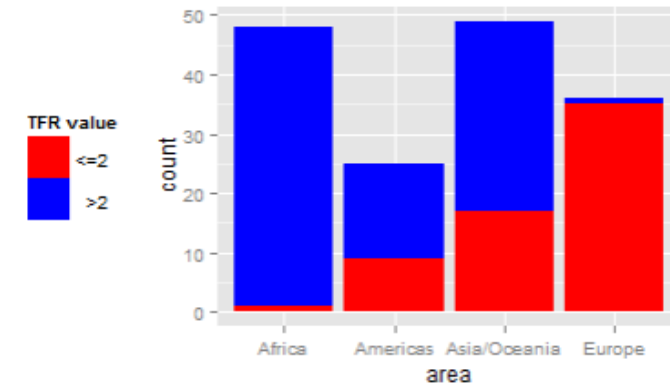


Theme: Titles, Tick Marks, and Tick Labels – ggplot2

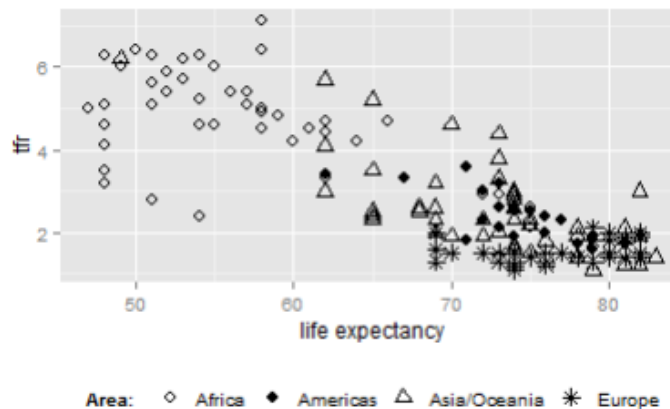
```
theme(title=element_text(color="blue", size=30),  
      axis.title=element_text(size=14, face="bold"),  
      axis.title.x=element_text(color="green"),  
      axis.text=element_text(size=14),  
      axis.text.y=element_text(color="black"),  
      axis.text.x=element_text(color="purple"),  
      axis.ticks.y=element_blank())
```



Theme: Legends – ggplot2



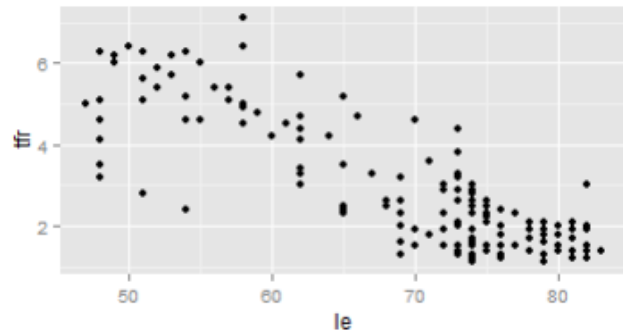
```
p + geom_bar() +
  scale_fill_manual(name="TFR value",
    values = c("red", "blue"),
    labels=c("<=2", ">2")) +
  theme(legend.position="left",
    legend.text.align=1)
```



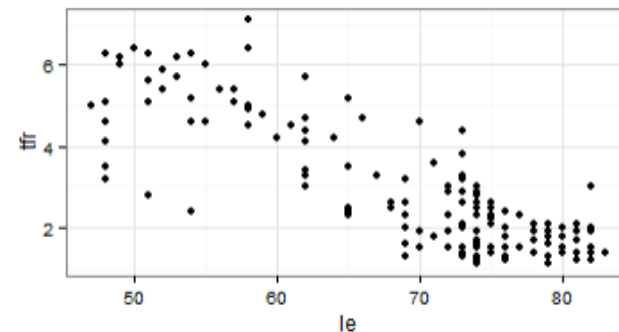
```
p + geom_point(aes(x=le, y=tfr,
  shape=area, fill=NULL), size = 3) +
  xlab("life expectancy") +
  scale_shape_manual(name="Area: ",
    values=c(1,16,2,8)) +
  theme(legend.key=element_blank(),
    legend.direction="horizontal",
    legend.position="bottom")
```

Theme: Overall Look– ggplot2

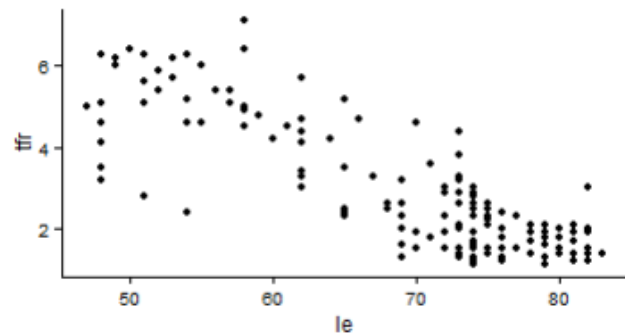
```
p + geom_point() + theme_gray()
```



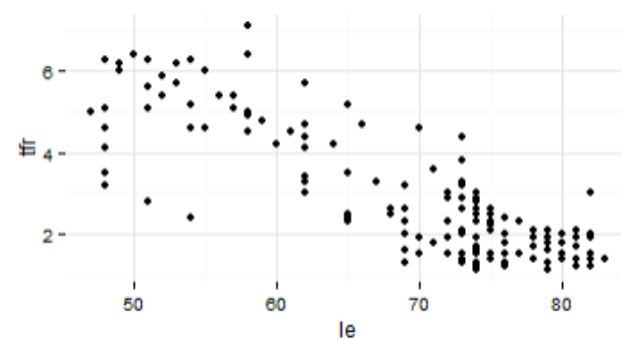
```
p + geom_point() + theme_bw()
```



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

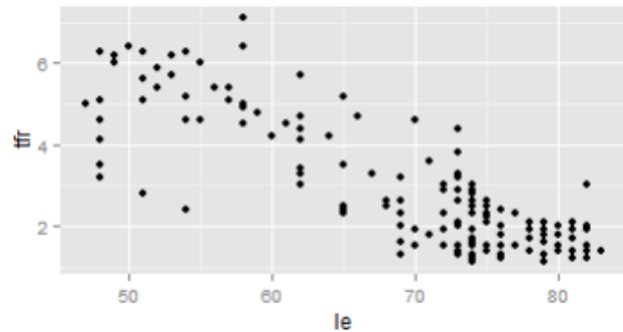


```
p + geom_point() + theme_minimal()
```

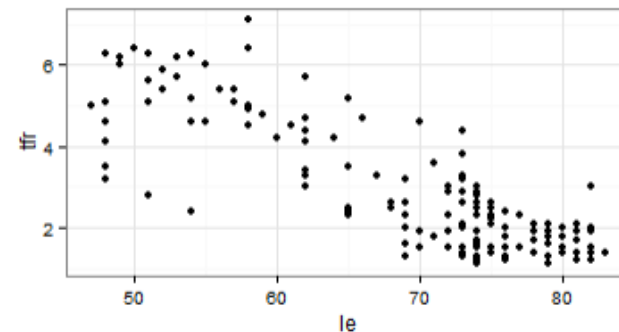


Theme: Overall Look– ggplot2

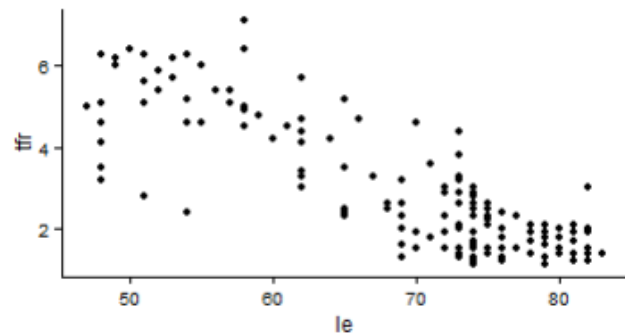
```
p + geom_point() + theme_gray()
```



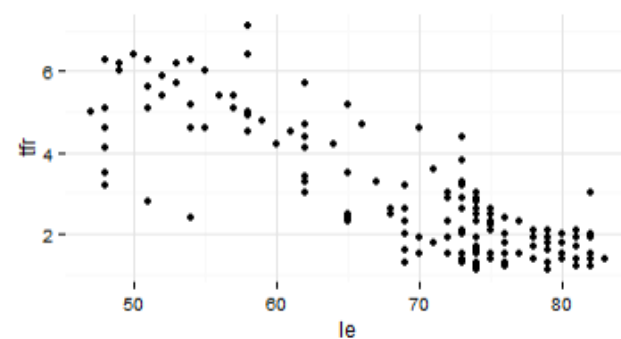
```
p + geom_point() + theme_bw()
```



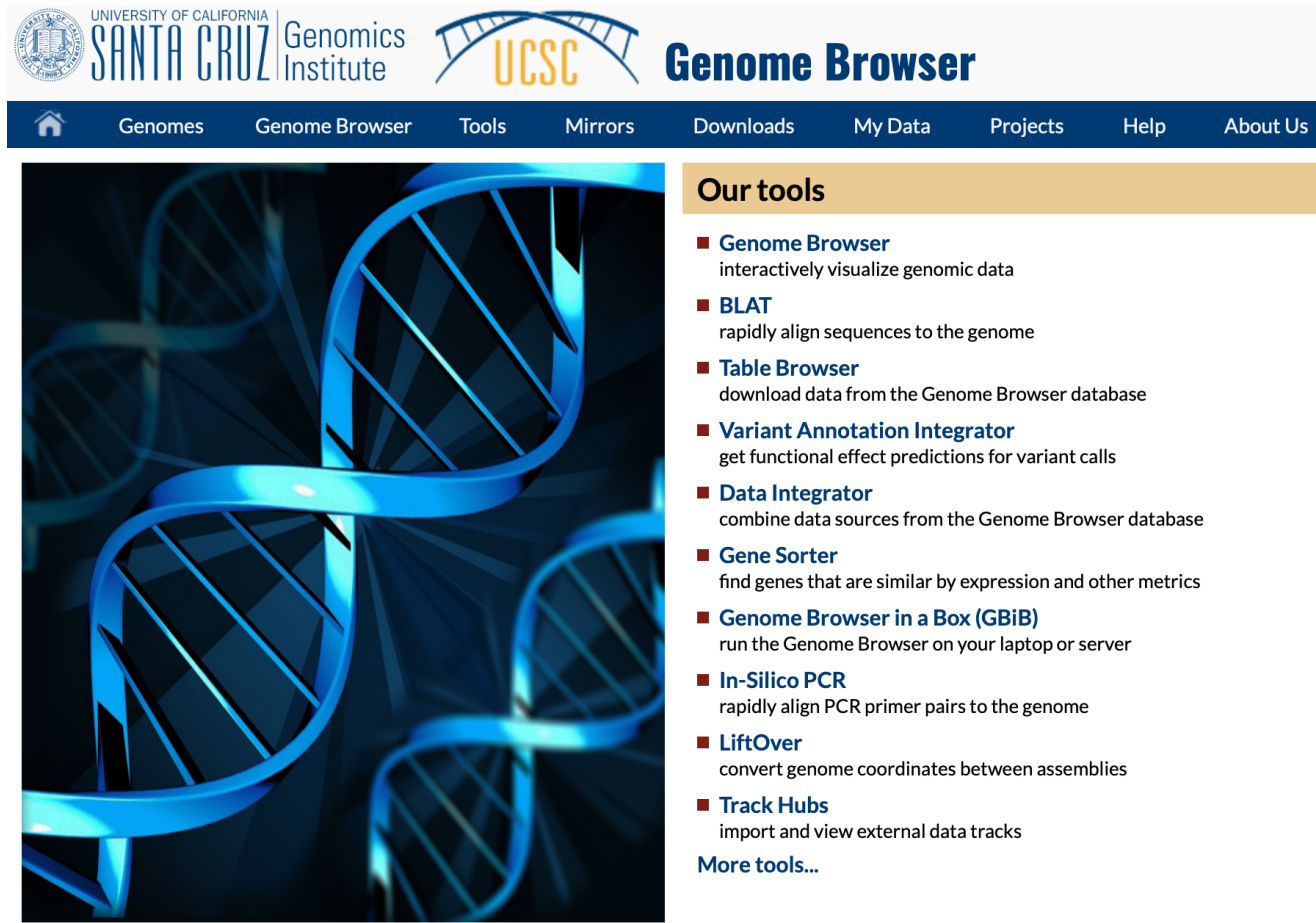
```
p + geom_point() + theme_classic()
```



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



Genome browser



UNIVERSITY OF CALIFORNIA
SANTA CRUZ Genomics
Institute

UCSC

Genome Browser

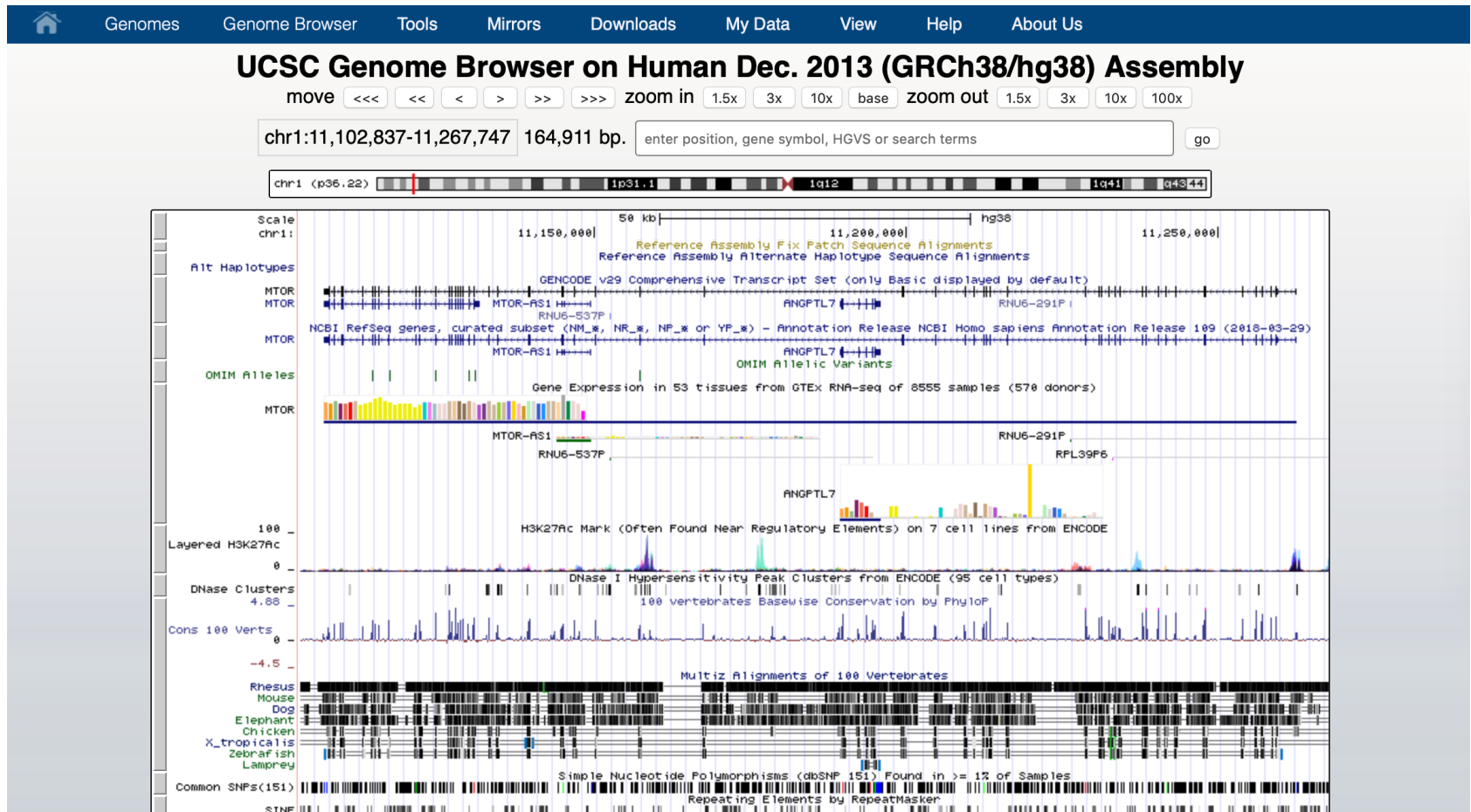
Home Genomes Genome Browser Tools Mirrors Downloads My Data Projects Help About Us

Our tools

- **Genome Browser**
interactively visualize genomic data
- **BLAT**
rapidly align sequences to the genome
- **Table Browser**
download data from the Genome Browser database
- **Variant Annotation Integrator**
get functional effect predictions for variant calls
- **Data Integrator**
combine data sources from the Genome Browser database
- **Gene Sorter**
find genes that are similar by expression and other metrics
- **Genome Browser in a Box (GBiB)**
run the Genome Browser on your laptop or server
- **In-Silico PCR**
rapidly align PCR primer pairs to the genome
- **LiftOver**
convert genome coordinates between assemblies
- **Track Hubs**
import and view external data tracks

[More tools...](#)

Genome browser



Available genomes -Genome browser

Genomes [\[hide\]](#)

great apes	human, baboon, bonobo, chimp, gibbon, gorilla, orangutan
non-ape primates	bushbaby, marmoset, mouse lemur, rhesus macaque, squirrel monkey, tarsier, tree shrew
non-primate mammals	mouse, alpaca, armadillo, cat, Chinese hamster, cow, dog, dolphin, elephant, ferret, guinea pig, hedgehog, horse, kangaroo rat, manatee, Minke whale, naked mole-rat, opossum, panda, pig, pika, platypus, rabbit, rat, rock hyrax, sheep, shrew, sloth, squirrel, Tasmanian devil, tenrec, wallaby, white rhinoceros
non-mammal chordates	American alligator, Atlantic cod, budgerigar, chicken, coelacanth, elephant shark, Fugu, lamprey, lizard, medaka, medium ground finch, Nile tilapia, painted turtle, stickleback, Tetraodon, turkey, <i>Xenopus tropicalis</i> , zebra finch, zebrafish
invertebrates	<i>Caenorhabditis</i> spp (5), <i>Drosophila</i> spp. (11), Ebola virus, honey bee, lancelet, mosquito, <i>P. Pacificus</i> , sea hare, sea squirt, sea urchin, yeast

Available information - Genome browser

track search default tracks default order hide all add custom tracks track hubs configure multi-region reverse resize refresh

collapse all Use drop-down controls below and press refresh to alter tracks displayed. Tracks with lots of items will automatically be displayed in more compact modes. expand all

Mapping and Sequencing

refresh

Base Position dense ▾	P12 Fix Patches pack ▾	P12 Alt Haplotypes dense ▾	P12 Assembly hide ▾	Centromeres hide ▾	P12 Chromosome Band hide ▾
Clone Ends hide ▾	18 FISH Clones hide ▾	P12 Gap hide ▾	P12 GC Percent hide ▾	GRC Contigs hide ▾	GRC Incident hide ▾
Hg19 Diff hide ▾	P12 INSDC hide ▾	LRG Regions hide ▾	Mappability... hide ▾	P12 RefSeq Acc hide ▾	Restr Enzymes hide ▾
Scaffolds hide ▾	Short Match hide ▾	STS Markers hide ▾			

Genes and Gene Predictions

refresh

P12 GENCODE v29 pack ▾	NCBI RefSeq pack ▾	P12 Other RefSeq hide ▾	P12 Updated All GENCODE... hide ▾	P12 AUGUSTUS hide ▾	CCDS hide ▾
Geneid Genes hide ▾	P12 Genscan Genes hide ▾	19 IKMC Genes Mapped hide ▾	LRG Transcripts hide ▾	MANE select v0.5 hide ▾	P12 MGC Genes hide ▾
Non-coding RNA... hide ▾	Old UCSC Genes hide ▾	P12 ORFeome Clones hide ▾	P12 Pfam in UCSC Gene hide ▾	RetroGenes V9 hide ▾	SGP Genes hide ▾
SIB Genes hide ▾	New TransMap V5... hide ▾	P12 UCSC Alt Events hide ▾	UniProt hide ▾		

Add custom tracks - Genome browser

Add Custom Tracks

clade genome assembly

Display your own data as custom annotation tracks in the browser. Data must be formatted in [bigBed](#), [bigBarChart](#), [bigChain](#), [bigGenePred](#), [bigInteract](#), [bigMaf](#), [bigPsl](#), [bigWig](#), [BAM](#), [barChart](#), [VCF](#), [BED](#), [BED detail](#), [bedGraph](#), [broadPeak](#), [CRAM](#), [GFF](#), [GTF](#), [interact](#), [MAF](#), [narrowPeak](#), [Personal Genome SNP](#), [PSL](#), or [WIG](#) formats. To configure the display, set [track](#) and [browser](#) line attributes as described in the [User's Guide](#). Data in the bigBed, bigWig, bigGenePred, BAM and VCF formats can be provided via only a URL or embedded in a track line in the box below. Examples are [here](#). If you do not have web-accessible data storage available, please see the [Hosting](#) section of the Track Hub Help documentation.

Please note a much more efficient way to load data is to use [Track Hubs](#), which are loaded from the [Track Hubs Portal](#) found in the menu under My Data.

Paste URLs or data:

Or upload:

Optional track documentation: Or upload:

In-Slico PCR - Genome browser

[Home](#) [Genomes](#) [Genome Browser](#) [Tools](#) [Mirrors](#) [Downloads](#) [My Data](#) [Projects](#) [Help](#) [About Us](#)

UCSC In-Silico PCR

Genome:
Human

Assembly:
Dec. 2013 (GRCh38/hg38)

Target:
genome assembly

Forward Primer:

Reverse Primer:

submit

Max Product Size: 4000

Min Perfect Match: 15

Min Good Match: 15

Flip Reverse Primer: ☐

About In-Silico PCR

In-Silico PCR searches a sequence database with a pair of PCR primers, using an indexing strategy for fast performance. See an example [video](#) on our YouTube channel.

Configuration Options

Genome and Assembly - The sequence database to search.
Target - If available, choose to query transcribed sequences.
Forward Primer - Must be at least 15 bases in length.
Reverse Primer - On the opposite strand from the forward primer. Minimum length of 15 bases.
Max Product Size - Maximum size of amplified region.
Min Perfect Match - Number of bases that match exactly on 3' end of primers. Minimum match size is 15.
Min Good Match - Number of bases on 3' end of primers where at least 2 out of 3 bases match.
Flip Reverse Primer - Invert the sequence order of the reverse primer and complement it.



Integrative Genomics Viewer

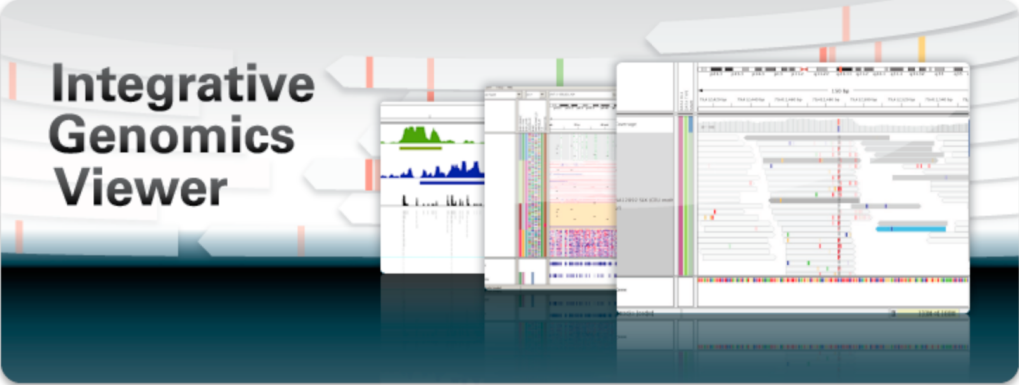
- Home
- Downloads
- Documents
 - IGV User Guide
 - Tutorial Videos
 - File Formats
 - Hosted Genomes
 - FAQ
- Release Notes
- Credits
- Contact

Search website

search

© 2013-2018
Broad Institute
and the Regents of the
University of California

Home



Overview



The **Integrative Genomics Viewer (IGV)** is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations.

IGV is available in multiple forms, including:

- the original **IGV** - a Java desktop application,
- IGV-Web** - a web application,
- igv.js** - a JavaScript component that can be embedded in web pages (*for developers*)

This site is focused on the IGV desktop application. See <https://igv.org> for links to all forms of IGV.

Citing IGV

To cite your use of IGV in your publication, please reference one or more of:

James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, Jill P. Mesirov. [Integrative Genomics Viewer](#). *Nature Biotechnology* 29, 24–26 (2011). (Free PMC article [here](#)).

Helga Thorvaldsdóttir, James T. Robinson, Jill P. Mesirov. [Integrative Genomics Viewer \(IGV\): high-performance genomics data visualization and exploration](#). *Briefings in Bioinformatics* 14, 178–192 (2013).

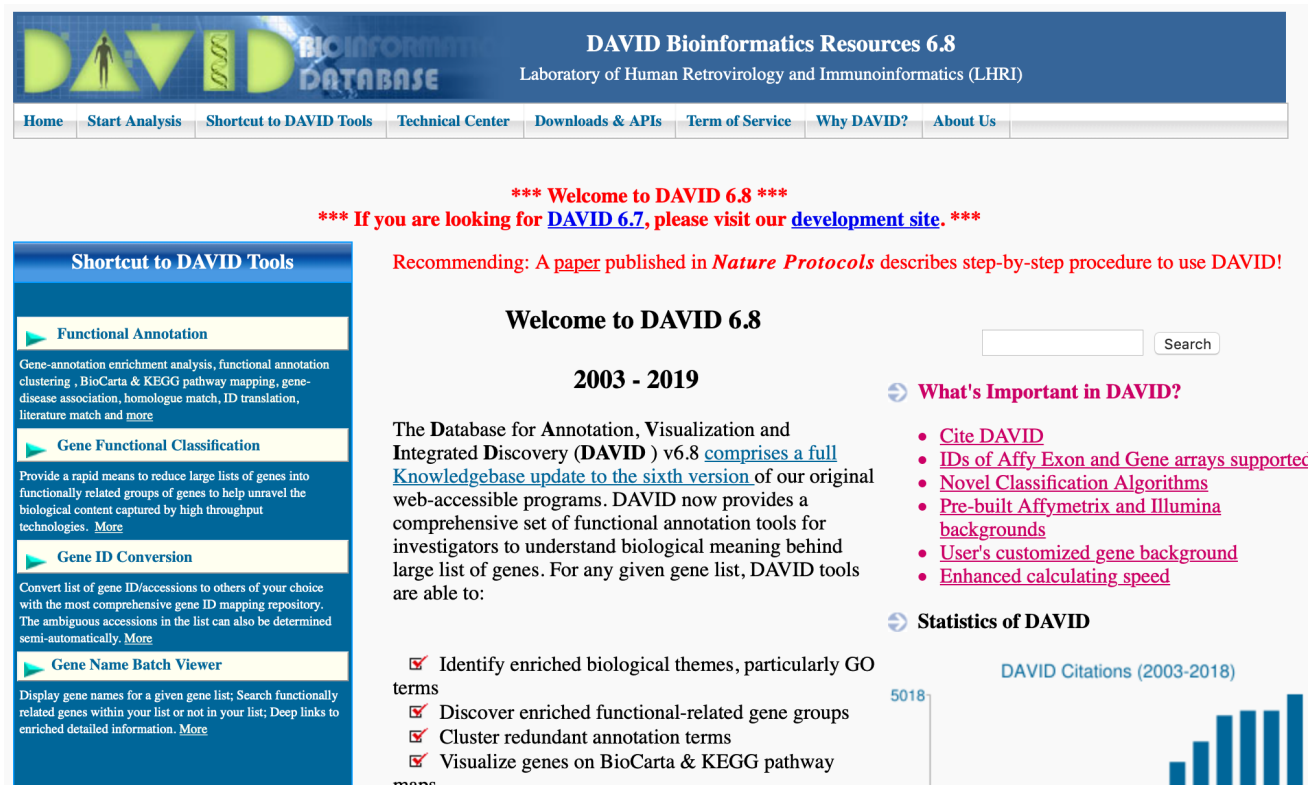
James T. Robinson, Helga Thorvaldsdóttir, Aaron M. Wenger, Ahmet Zehir, Jill P. Mesirov. [Variant Review with the Integrative Genomics Viewer \(IGV\)](#). *Cancer Research* 77(21), 31–34 (2017).



Practice - IGV

- Path - /var2/users/Practice/IGV
- Files
 - H1.Myc.bed
 - H1.Nanog.bed
 - GSM438363_UCSD.IMR90.mRNA-Seq.mRNA-seq_imr90_r1.wig

DAVID



DAVID Bioinformatics Resources 6.8
Laboratory of Human Retrovirology and Immunoinformatics (LHRI)

Home | Start Analysis | Shortcut to DAVID Tools | Technical Center | Downloads & APIs | Term of Service | Why DAVID? | About Us

*** Welcome to DAVID 6.8 ***
*** If you are looking for [DAVID 6.7](#), please visit our [development site](#). ***

Shortcut to DAVID Tools

- Functional Annotation**
Gene-annotation enrichment analysis, functional annotation clustering, BioCarta & KEGG pathway mapping, gene-disease association, homologue match, ID translation, literature match and [more](#)
- Gene Functional Classification**
Provide a rapid means to reduce large lists of genes into functionally related groups of genes to help unravel the biological content captured by high throughput technologies. [More](#)
- Gene ID Conversion**
Convert list of gene ID/accessions to others of your choice with the most comprehensive gene ID mapping repository. The ambiguous accessions in the list can also be determined semi-automatically. [More](#)
- Gene Name Batch Viewer**
Display gene names for a given gene list; Search functionally related genes within your list or not in your list; Deep links to enriched detailed information. [More](#)

Welcome to DAVID 6.8

2003 - 2019

The Database for Annotation, Visualization and Integrated Discovery (DAVID) v6.8 [comprises a full Knowledgebase update to the sixth version](#) of our original web-accessible programs. DAVID now provides a comprehensive set of functional annotation tools for investigators to understand biological meaning behind large list of genes. For any given gene list, DAVID tools are able to:

- Identify enriched biological themes, particularly GO terms
- Discover enriched functional-related gene groups
- Cluster redundant annotation terms
- Visualize genes on BioCarta & KEGG pathway maps

What's Important in DAVID?

- [Cite DAVID](#)
- [IDs of Affy Exon and Gene arrays supported](#)
- [Novel Classification Algorithms](#)
- [Pre-built Affymetrix and Illumina backgrounds](#)
- [User's customized gene background](#)
- [Enhanced calculating speed](#)

Statistics of DAVID

DAVID Citations (2003-2018)



Year	Citations
2003	~100
2004	~150
2005	~200
2006	~250
2007	~300
2008	~350
2009	~400
2010	~450
2011	~500
2012	~550
2013	~600
2014	~650
2015	~700
2016	~750
2017	~800
2018	~850

DAVID is a tool for Gene ontology analysis to understand biological meaning behind interest gene list

Insert gene list - DAVID

1. 유전자 리스트 삽입

The screenshot shows the DAVID web interface. On the left, the 'Upload Gene List' sidebar contains the following elements:

- Links: [Demolist](#), [Demolist 2](#), [Upload Help](#)
- Step 1: Enter Gene List
 - A: Paste a list (with a text input field and a 'Clear' button)
 - Or
 - B: Choose From a File (with '파일 선택' and '선택한 파일 없음' buttons)
 - ☐ Multi-List File ?
- Step 2: Select Identifier (with a dropdown menu showing 'AFFYMETRIX_3PRIME_IVT_ID')
- Step 3: List Type (with radio buttons for 'Gene List' and 'Background')
- Step 4: Submit List (with a 'Submit List' button)

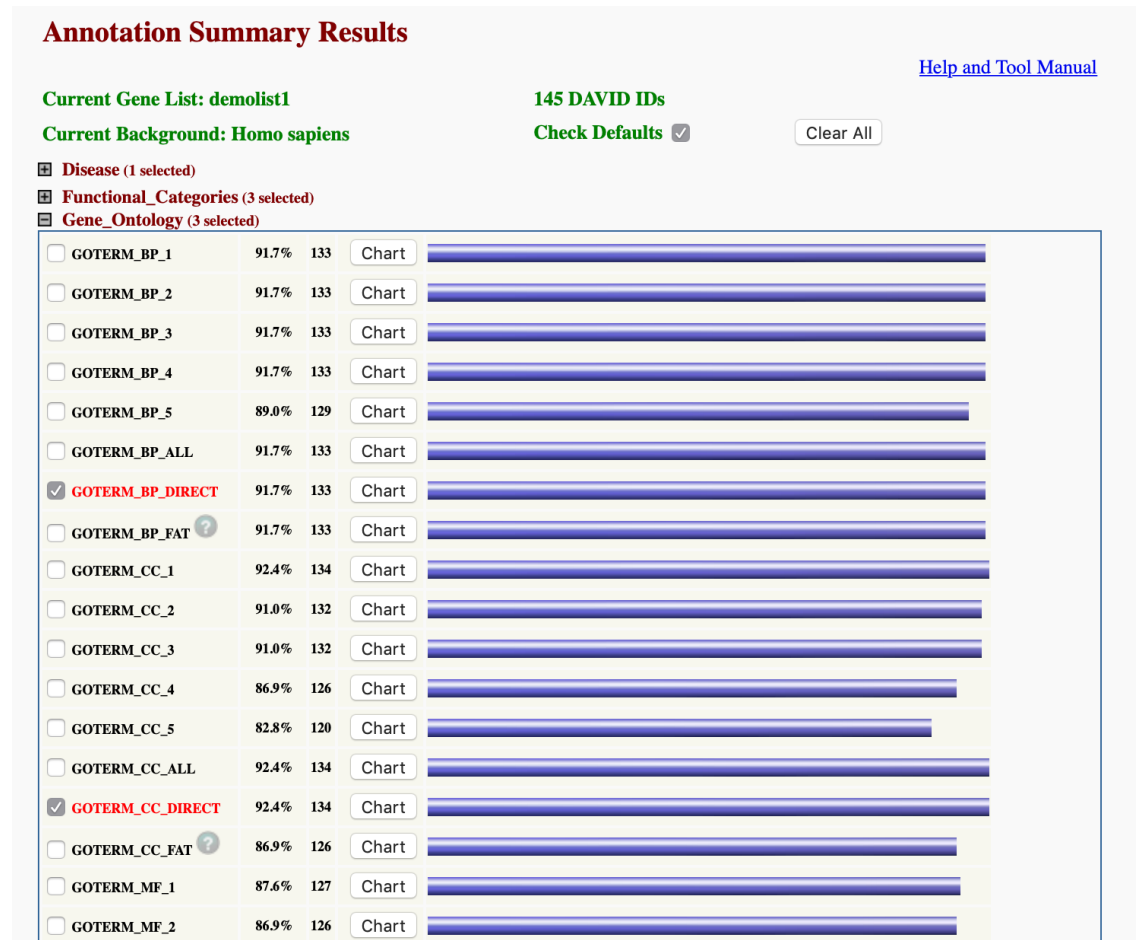
On the right, the 'Annotation Summary Results' panel displays:

- Current Gene List:** demolist1
- Current Background:** Homo sapiens
- 145 DAVID IDs**
- Check Defaults** (checked)
- Clear All** button
- Annotation categories with counts:
 - Disease (1 selected)
 - Functional_Categories (3 selected)
 - Gene_Ontology (3 selected)
 - General_Annotations (0 selected)
 - Literature (0 selected)
 - Main_Accessions (0 selected)
 - Pathways (3 selected)
 - Protein_Domains (3 selected)
 - Protein_Interactions (0 selected)
 - Tissue_Expression (0 selected)
- ***Red annotation categories denote DAVID defined defaults***
- Combined View for Selected Annotation** (with buttons for Functional Annotation Clustering, Functional Annotation Chart, and Functional Annotation Table)

Three black arrows point from Korean text labels to specific parts of the interface:

- Arrow 1 points to the 'A: Paste a list' text input field.
- Arrow 2 points to the 'AFFYMETRIX_3PRIME_IVT_ID' dropdown menu.
- Arrow 3 points to the 'Gene List' radio button.

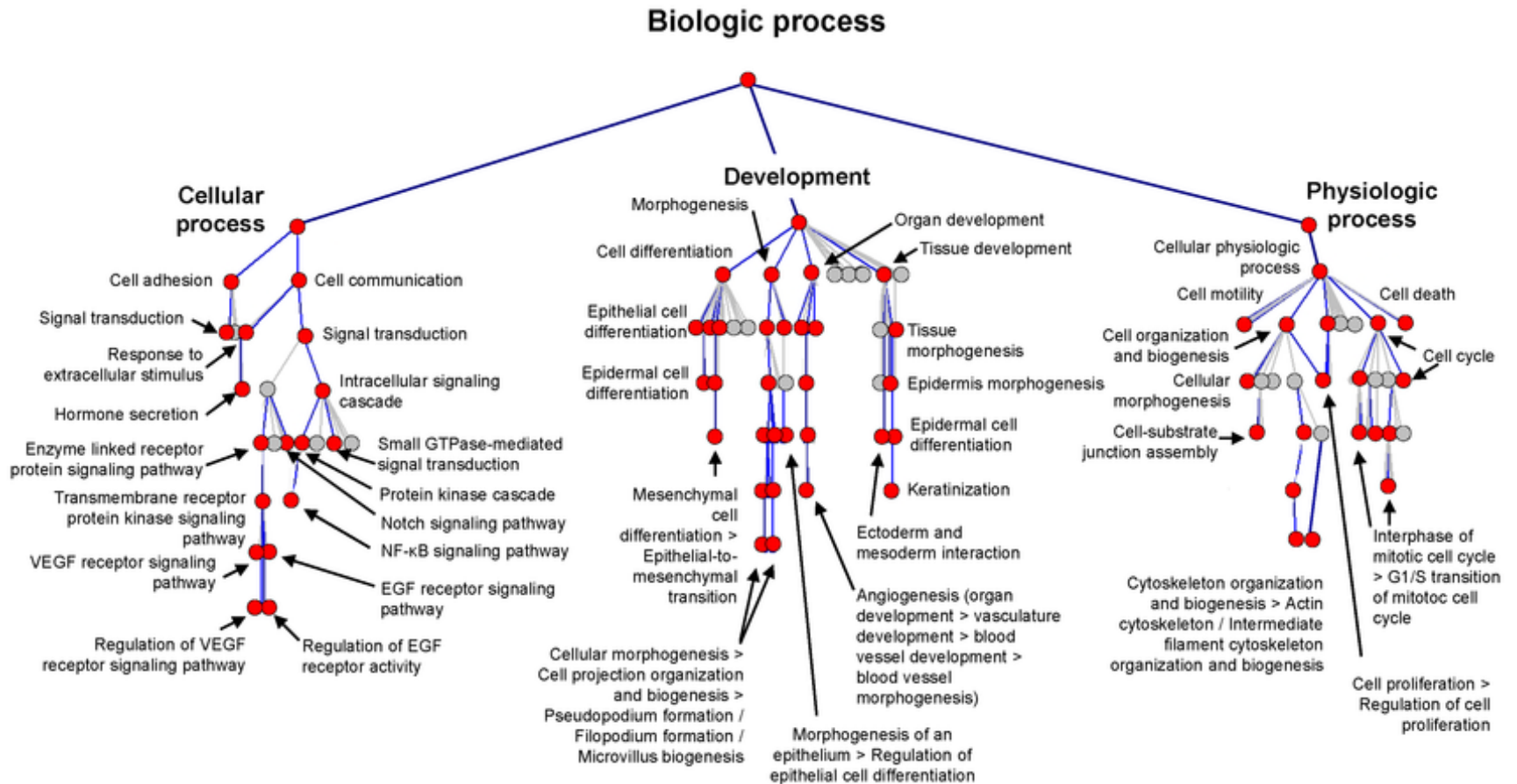
Gene ontology - DAVID



Biological process - DAVID

Sublist	Category	Term	RT	Genes	Count	%	P-Value	Benjamini
☐	GOTERM_BP_3	disruption of cells of other organism	RT		7	4.8	1.1E-7	7.9E-5
☐	GOTERM_BP_3	response to fungus	RT		7	4.8	3.7E-6	1.3E-3
☐	GOTERM_BP_3	regulation of cell death	RT		29	20.0	1.6E-5	3.7E-3
☐	GOTERM_BP_3	negative regulation of cell death	RT		21	14.5	2.6E-5	4.5E-3
☐	GOTERM_BP_3	response to bacterium	RT		16	11.0	2.7E-5	3.8E-3
☐	GOTERM_BP_3	modification of morphology or physiology of other organism	RT		8	5.5	3.0E-5	3.5E-3
☐	GOTERM_BP_3	positive regulation of response to stimulus	RT		34	23.4	3.7E-5	3.7E-3
☐	GOTERM_BP_3	regulation of cell proliferation	RT		28	19.3	7.2E-5	6.3E-3
☐	GOTERM_BP_3	antimicrobial humoral response	RT		6	4.1	1.1E-4	8.6E-3
☐	GOTERM_BP_3	defense response	RT		27	18.6	1.3E-4	9.1E-3
☐	GOTERM_BP_3	humoral immune response	RT		10	6.9	1.6E-4	1.0E-2
☐	GOTERM_BP_3	response to external biotic stimulus	RT		18	12.4	3.0E-4	1.7E-2
☐	GOTERM_BP_3	response to drug	RT		12	8.3	4.3E-4	2.3E-2
☐	GOTERM_BP_3	positive regulation of signaling	RT		26	17.9	5.1E-4	2.5E-2
☐	GOTERM_BP_3	defense response to other organism	RT		10	6.9	5.0E-4	2.6E-2

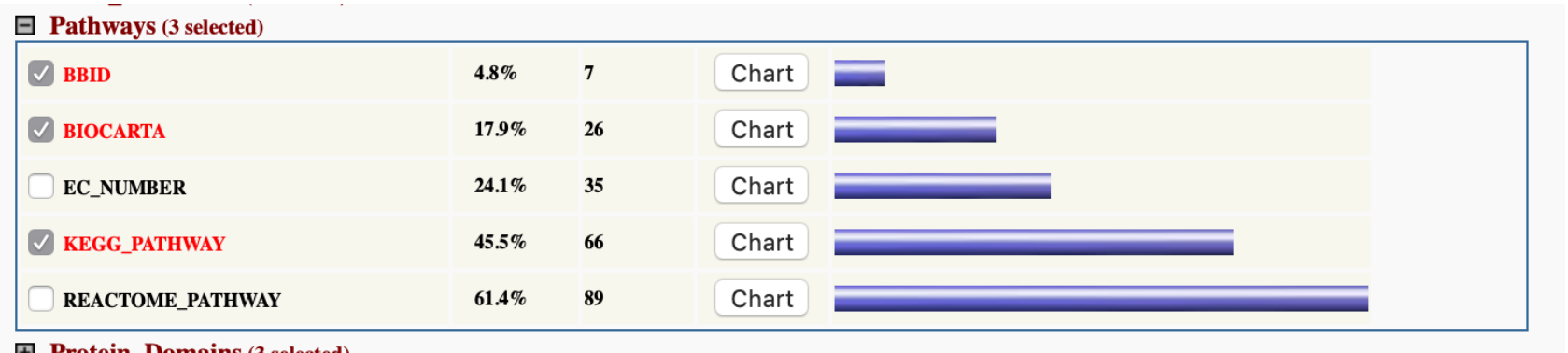
Structure of gene ontology - DAVID



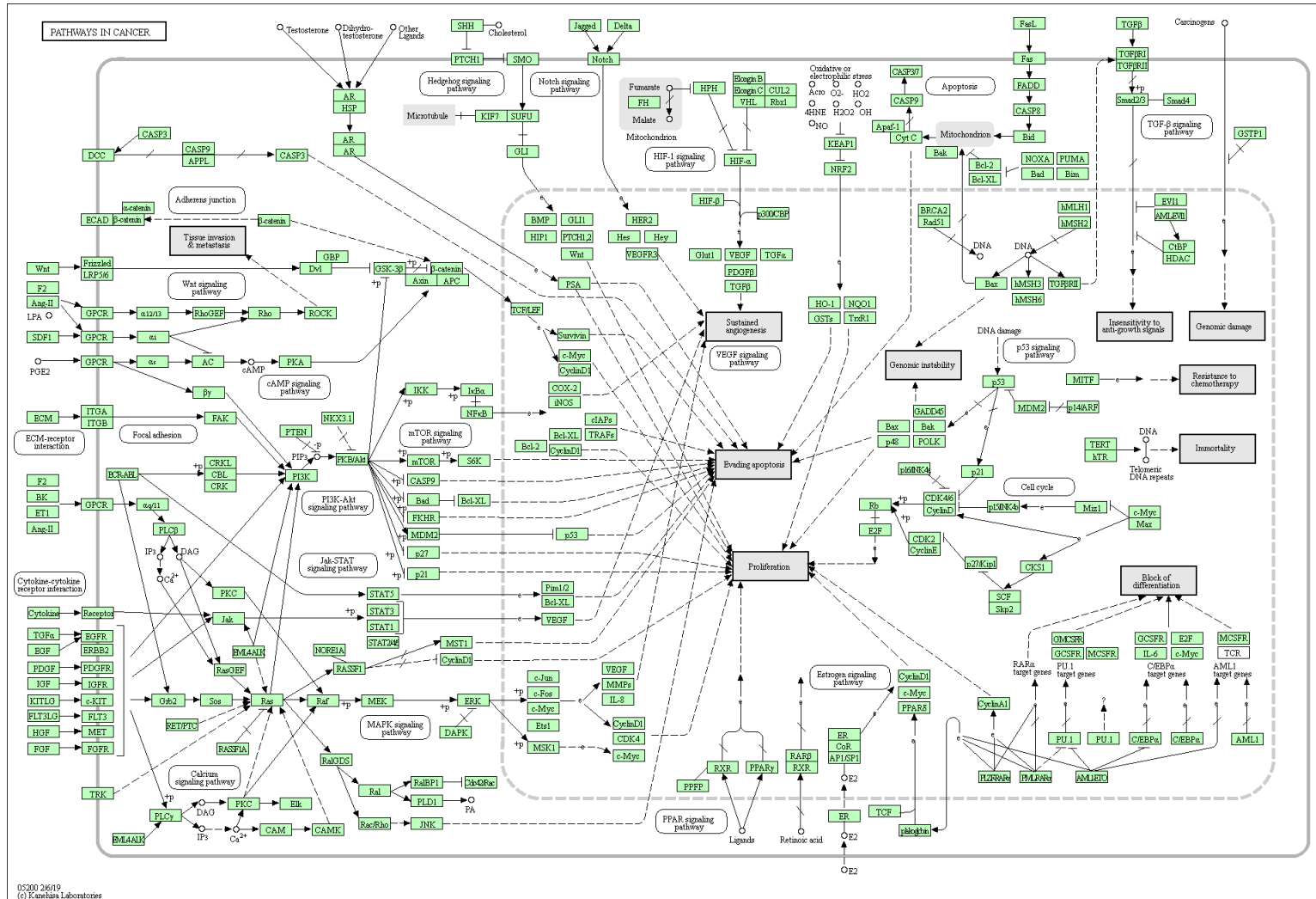
KEGG pathway - DAVID

Sublist	Category	Term	RT	Genes	Count	%	P-Value	Benjamini
☐	KEGG_PATHWAY	Pathways in cancer	RT		10	6.9	1.1E-2	8.6E-1
☐	KEGG_PATHWAY	Transcriptional misregulation in cancer	RT		6	4.1	2.0E-2	8.3E-1
☐	KEGG_PATHWAY	Calcium signaling pathway	RT		6	4.1	2.7E-2	7.9E-1
☐	KEGG_PATHWAY	MAPK signaling pathway	RT		7	4.8	3.2E-2	7.5E-1
☐	KEGG_PATHWAY	Signaling pathways regulating pluripotency of stem cells	RT		5	3.4	4.3E-2	7.8E-1
☐	KEGG_PATHWAY	Small cell lung cancer	RT		4	2.8	4.6E-2	7.5E-1
☐	KEGG_PATHWAY	Prostate cancer	RT		4	2.8	5.0E-2	7.2E-1
☐	KEGG_PATHWAY	HIF-1 signaling pathway	RT		4	2.8	6.2E-2	7.5E-1
☐	KEGG_PATHWAY	Amyotrophic lateral sclerosis (ALS)	RT		3	2.1	8.1E-2	8.1E-1
☐	KEGG_PATHWAY	Endometrial cancer	RT		3	2.1	8.6E-2	7.9E-1
☐	KEGG_PATHWAY	Basal cell carcinoma	RT		3	2.1	9.2E-2	7.8E-1

KEGG pathway - DAVID



KEGG pathway - DAVID



THANK YOU.

BMS