GSS Presentation: About Tidyverse

Kim Phan

January 29, 2021

Outline

- About Tidyverse
  - `dplyr` & `ggplot2`
- Exploratory Data Analysis
- More Detailed Examples

About Tidyverse

Hadley Wickham is known for his development of his “tidyverse” packages, which support a tidy data approach to import data, analysis, and modeling

About Tidyverse

- The core tidyverse includes the packages that we’re likely to use in everyday data analyses
    - **ggplot2** - creates graphics
    - **dplyr** - data manipulation
    - tidyr - forms data into a consistent form
    - readr - reads csv, tsv, fwf
    - purr - works with functions and vectors
    - tibble - data frame manipulation
    - stringr - functions designed for string manipulation
    - forcats - alleviates common problems with factors

About Tidyverse

- The core tidyverse includes the packages that we’re likely to use in everyday data analyses
    - **ggplot2** - creates graphics
    - **dplyr** - data manipulation

Exploratory Data Analysis

About `dplyr`

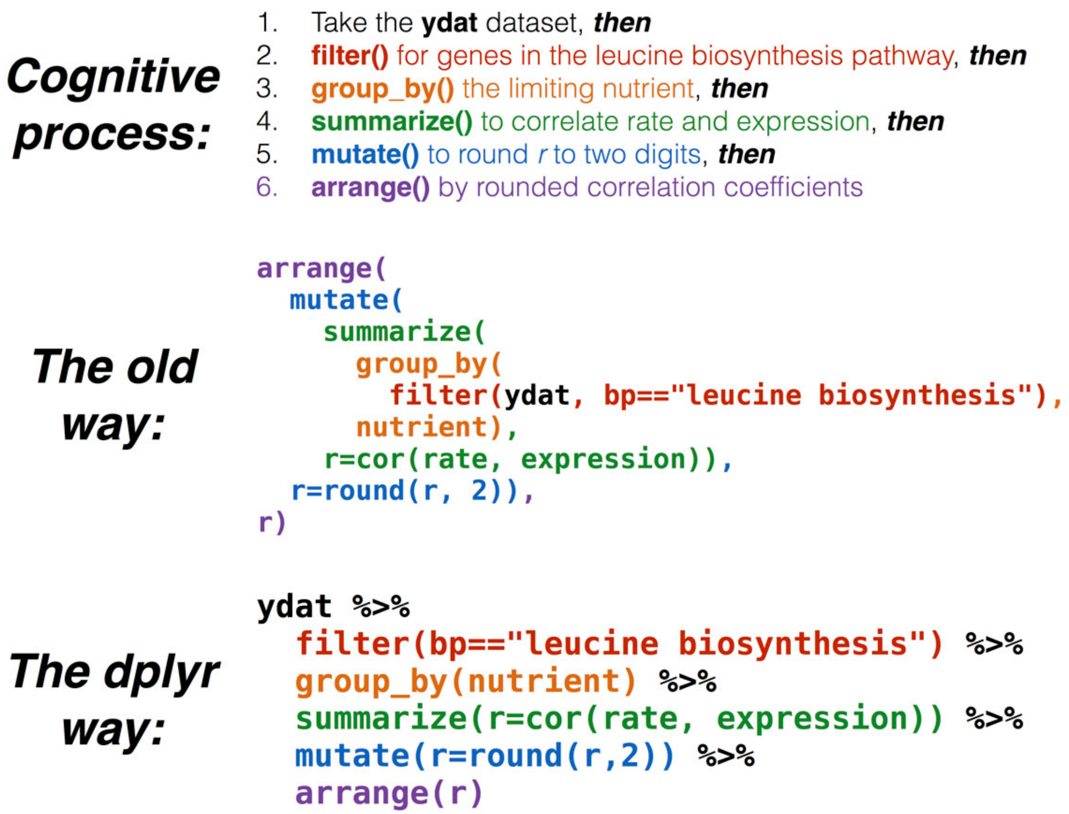
About `dplyr`

`dplyr` is used for data manipulation

About **dplyr**

`dplyr` is used for data manipulation

What is a “pipe”?



Cognitive process:

1. Take the **ydat** dataset, ***then***
2. **filter()** for genes in the leucine biosynthesis pathway, ***then***
3. **group_by()** the limiting nutrient, ***then***
4. **summarize()** to correlate rate and expression, ***then***
5. **mutate()** to round *r* to two digits, ***then***
6. **arrange()** by rounded correlation coefficients

The old way:

```
arrange(
  mutate(
    summarize(
      group_by(
        filter(ydat, bp=="leucine biosynthesis"),
      nutrient),
    r=cor(rate, expression)),
  r=round(r, 2)),
r)
```

The dplyr way:

```
ydat %>%
  filter(bp=="leucine biosynthesis") %>%
  group_by(nutrient) %>%
  summarize(r=cor(rate, expression)) %>%
  mutate(r=round(r,2)) %>%
  arrange(r)
```

About `dplyr`

`dplyr` is used for data manipulation

- `mutate()` adds new variables
- `select()` picks variables based on their names
- `filter()` picks cases based on their values
- `summarise()` reduces multiple values down to a single summary
- `arrange()` changes the ordering of the rows

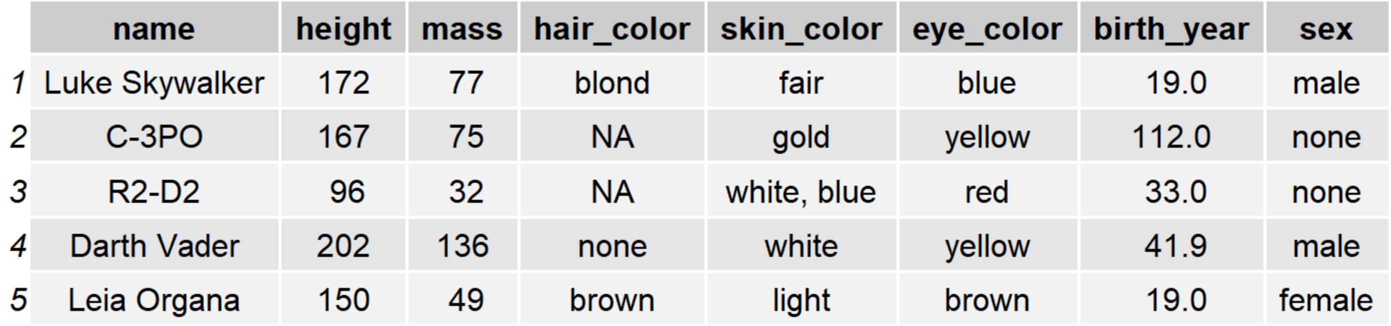
About `dplyr`

Data: `Star Wars` - A tibble with 87 rows and 13 variables:

- name: Name of character
- height: Height (cm)
- mass: Weight (kg)
- hair_color, skin_color, eye_color
- birth_year
- sex, gender
- etc

About `dplyr`

Star Wars Data



	name	height	mass	hair_color	skin_color	eye_color	birth_year	sex
1	Luke Skywalker	172	77	blond	fair	blue	19.0	male
2	C-3PO	167	75	NA	gold	yellow	112.0	none
3	R2-D2	96	32	NA	white, blue	red	33.0	none
4	Darth Vader	202	136	none	white	yellow	41.9	male
5	Leia Organa	150	49	brown	light	brown	19.0	female

About dplyr

Making a summary table of Homeworld

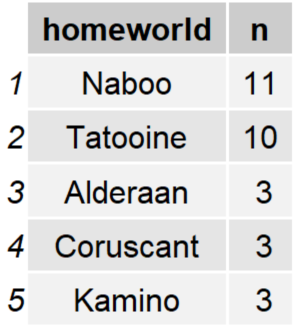
```
x1 <- starwars %>%
  select(homeworld) %>%
  group_by(homeworld) %>%
  na.omit() %>%
  tally %>%
  arrange(desc(n))

dim(x1)
```

```
## [1] 48  2
```

About `dplyr`

Making a summary table of the Top 5 common Homeworld

```
## Selecting by n
```

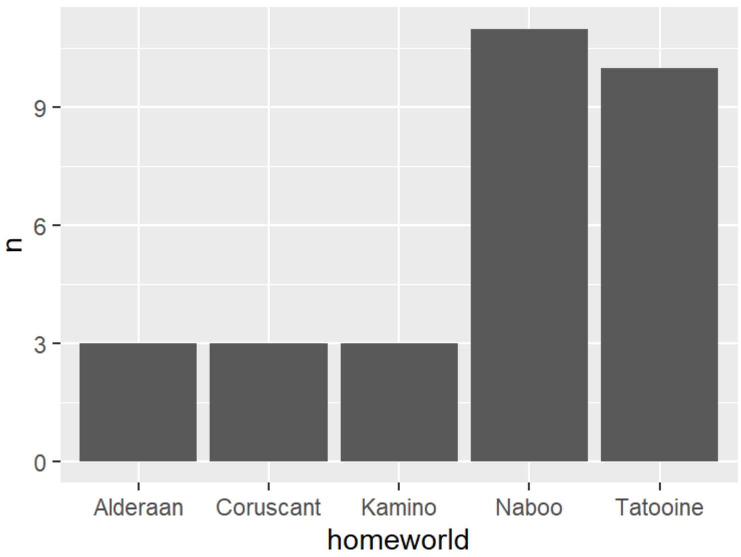


	homeworld	n
1	Naboo	11
2	Tatooine	10
3	Alderaan	3
4	Coruscant	3
5	Kamino	3

About **ggplot2**

Basic Plot of the Top 5 Homeworld of Characters in Star Wars

```
x1 %>%
  ggplot(mapping = aes(x = homeworld, y = n)) +
  geom_bar(stat = "identity")
```



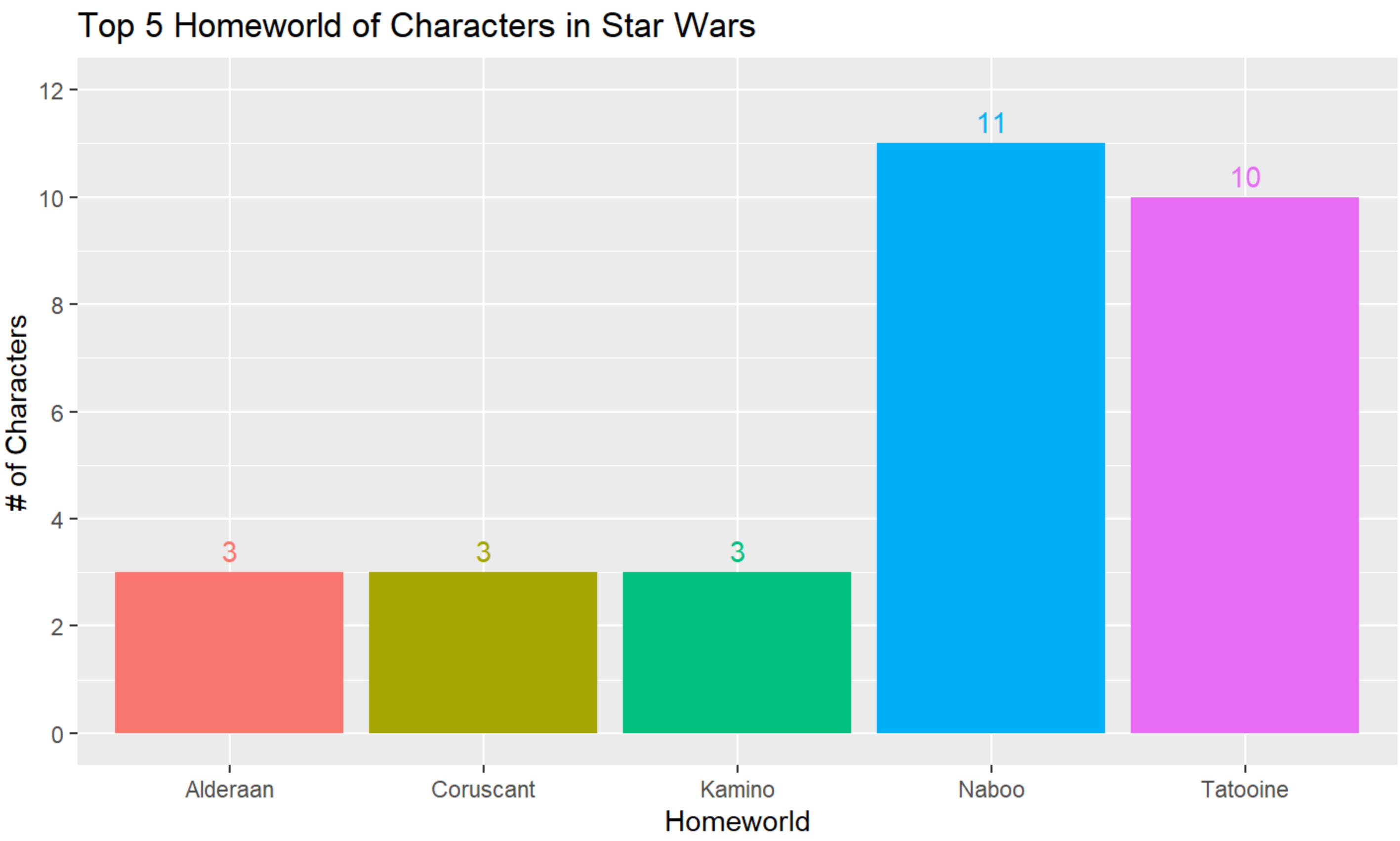
About `ggplot2`

Plotting the Top 5 Homeworld and adding aesthetics

```r
x1 %>%
  ggplot(mapping = aes(x = homeworld, y = n,
                       fill = homeworld)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n, color = homeworld),
            vjust=-0.5) +
  labs(x = "Homeworld", y = "# of Characters",
       title = "Top 5 Homeworld of Characters in Star Wars") +
  theme(legend.position = "") +
  scale_y_continuous(limits = c(0,12),
                     breaks = seq(0,12,2))
```

About `ggplot2`

Plotting the Top 5 Homeworld and adding aesthetics



About `dplyr`

Which 5 characters appeared in the most amount of movies?

```
starwars %>%
  select(films)
```

```
## # A tibble: 87 x 1
##    films
##    <list>
##  1 <chr [5]>
##  2 <chr [6]>
##  3 <chr [7]>
##  4 <chr [4]>
##  5 <chr [5]>
##  6 <chr [3]>
##  7 <chr [3]>
##  8 <chr [1]>
##  9 <chr [1]>
## 10 <chr [6]>
## # ... with 77 more rows
```

About `dplyr`

Which 5 characters appeared in the most amount of movies?

```
starwars %>%
  unnest(films) %>%
  select(films) %>%
  unique()
```

```
## # A tibble: 7 x 1
##   films
##   <chr>
## 1 The Empire Strikes Back
## 2 Revenge of the Sith
## 3 Return of the Jedi
## 4 A New Hope
## 5 The Force Awakens
## 6 Attack of the Clones
## 7 The Phantom Menace
```

About **dplyr**

Which 5 characters appeared in the most amount of movies?

```
starwars %>%
  unnest(films) %>%
  group_by(name) %>%
  tally %>%
  arrange(desc(n)) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   name               n
##   <chr>          <int>
## 1 R2-D2              7
## 2 C-3PO              6
## 3 Obi-Wan Kenobi     6
## 4 Chewbacca          5
## 5 Leia Organa        5
```

About **dplyr**

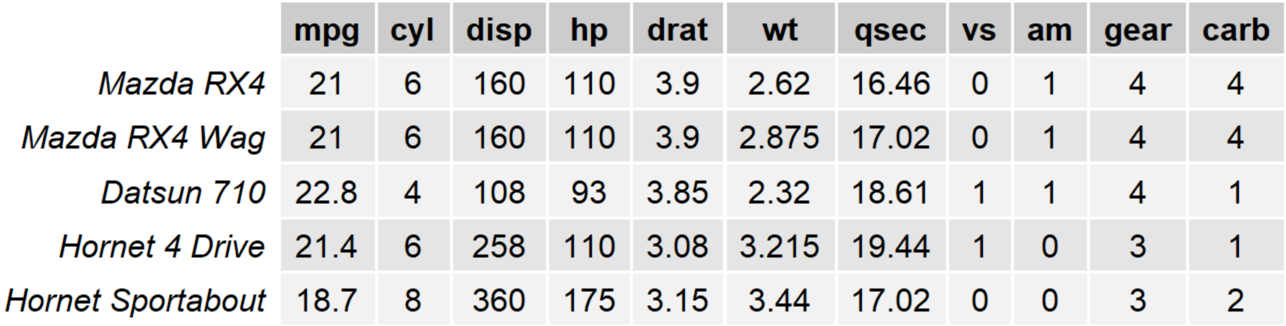
```
starwars %>%
  select(name:mass, gender, species) %>%
  mutate(
    type = case_when( height > 200 | mass > 200 ~ "large",
                      species == "Droid"        ~ "robot",
                      TRUE                      ~ "other"  )) %>%
  select(name, height, mass, species, type) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 5
##   name           height  mass species type
##   <chr>           <int> <dbl> <chr>   <chr>
## 1 Luke Skywalker    172    77 Human   other
## 2 C-3PO             167    75 Droid   robot
## 3 R2-D2              96    32 Droid   robot
## 4 Darth Vader       202   136 Human   large
## 5 Leia Organa       150    49 Human   other
```

About **ggplot2**

Let’s look into a different data set

```
data("mtcars")
```

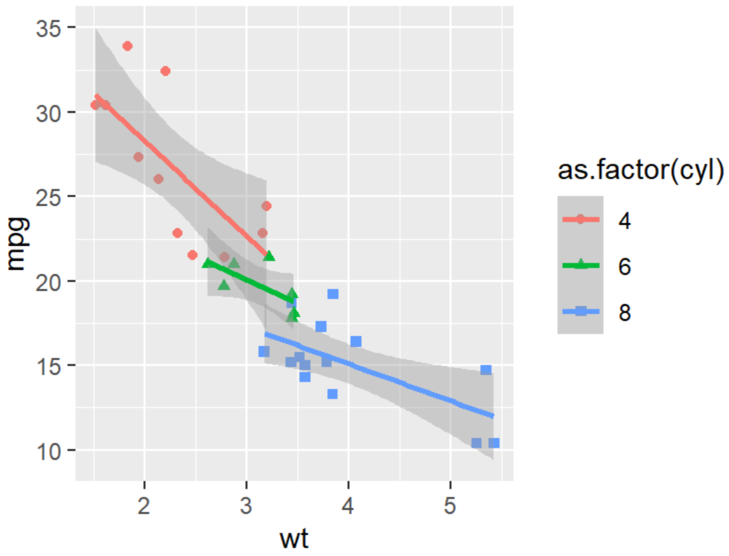


	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2

About **ggplot2**

```
# Add the regression line
ggplot(mtcars, aes(x=wt, y=mpg,
                   color=as.factor(cyl), shape=as.factor(cyl))) +
  geom_point() +
  geom_smooth(method=lm)
```

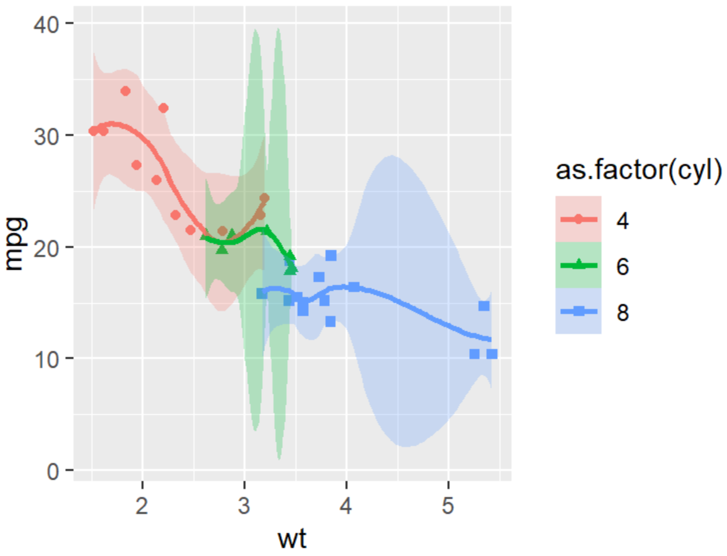
```
## `geom_smooth()` using formula 'y ~ x'
```



About **ggplot2**

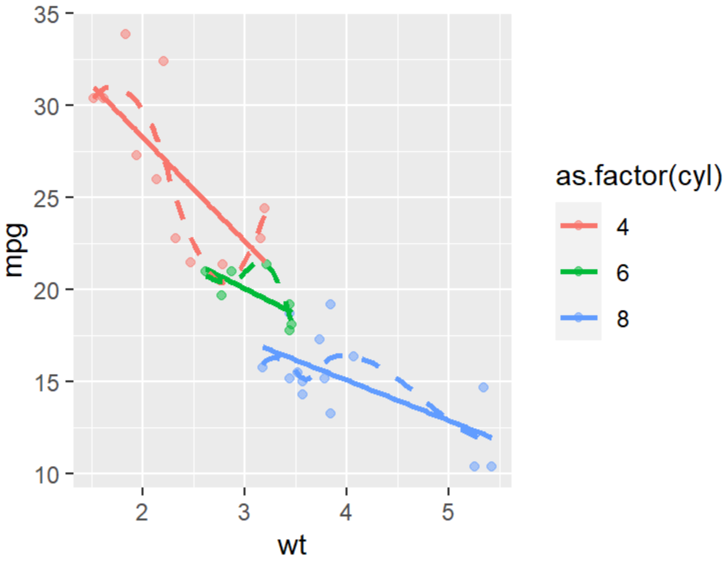
```r
# Loess method
ggplot(mtcars, aes(x=wt, y=mpg,
                   color = as.factor(cyl), shape = as.factor(cyl))) +
  geom_point() +
  geom_smooth(method=loess, alpha=0.25, aes(fill=as.factor(cyl)))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



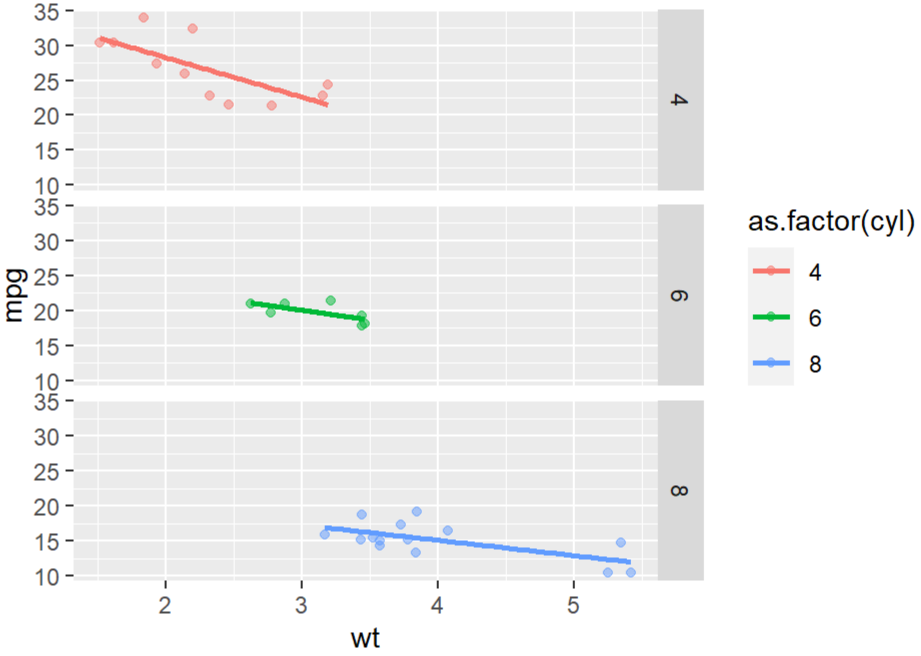
About **ggplot2**

```
ggplot(mtcars, aes(x=wt, y=mpg, color = as.factor(cyl))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method=lm, se=F) +
  geom_smooth(method=loess, se=F, linetype = "dashed")
```



About ggplot2

```
ggplot(mtcars, aes(x=wt, y=mpg, color = as.factor(cyl))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method=lm, se=F) +
  facet_grid(cyl~.)
```



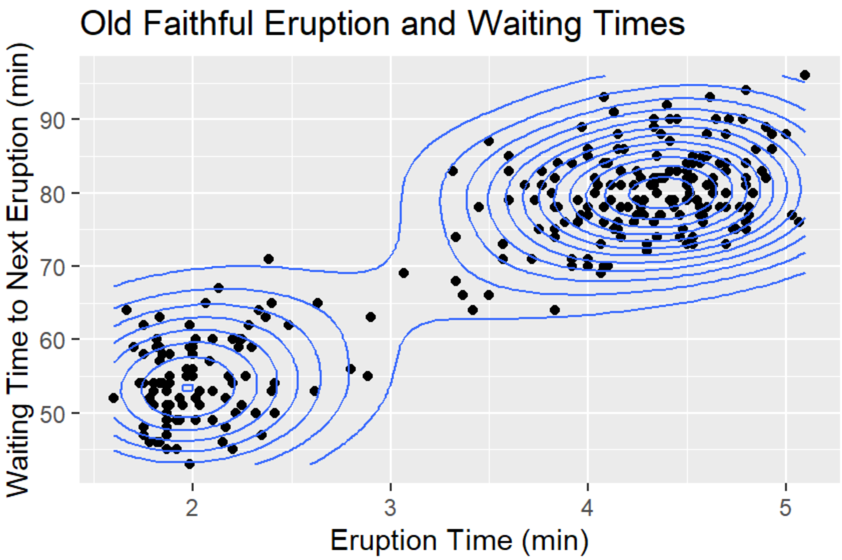
More Special Examples

More examples of ggplot2

About **ggplot2**

Countour Density Plot

```
ggplot(faithful, aes(x=eruptions, y=waiting)) +
  geom_point() +
  geom_density_2d() +
  labs(x="Eruption Time (min)", y="Waiting Time to Next Eruption (min)",
       title = "Old Faithful Eruption and Waiting Times")
```



About `ggplot2`

`ggfortify` lets `ggplot2` know how to interpret PCA objects. After loading `ggfortify`, we can use `ggplot2::autoplot` function for `stats::prcomp` and `stats::princomp` objects

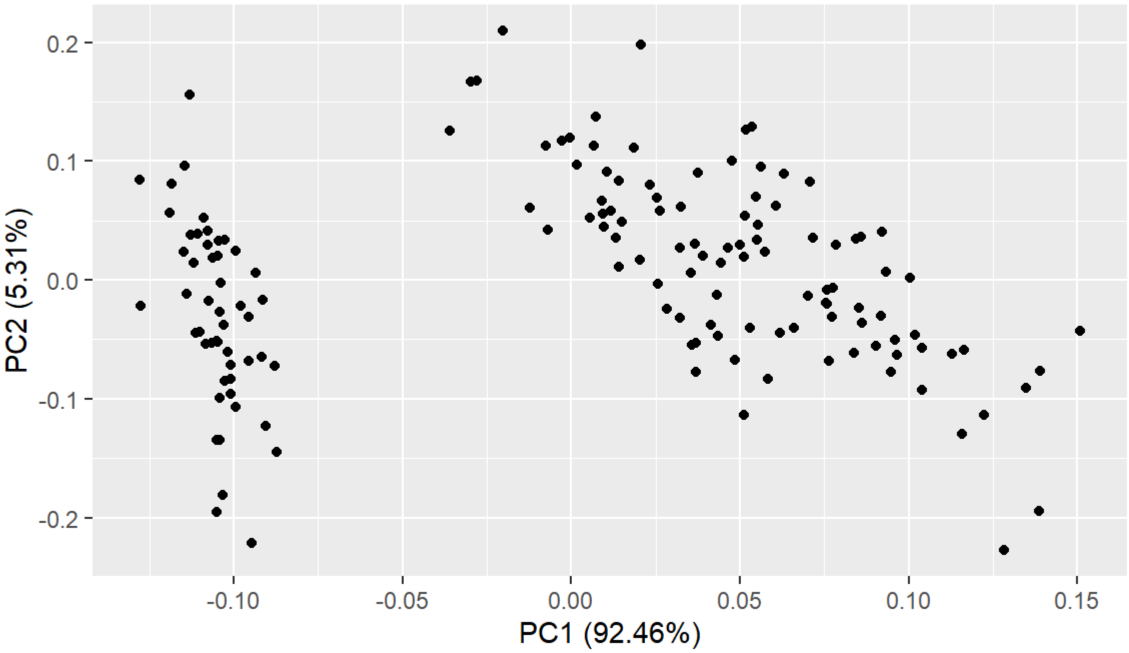
Plotting PCA results with `iris` data

```
library(ggfortify)
```

About ggplot2

Plotting PCA results with iris data

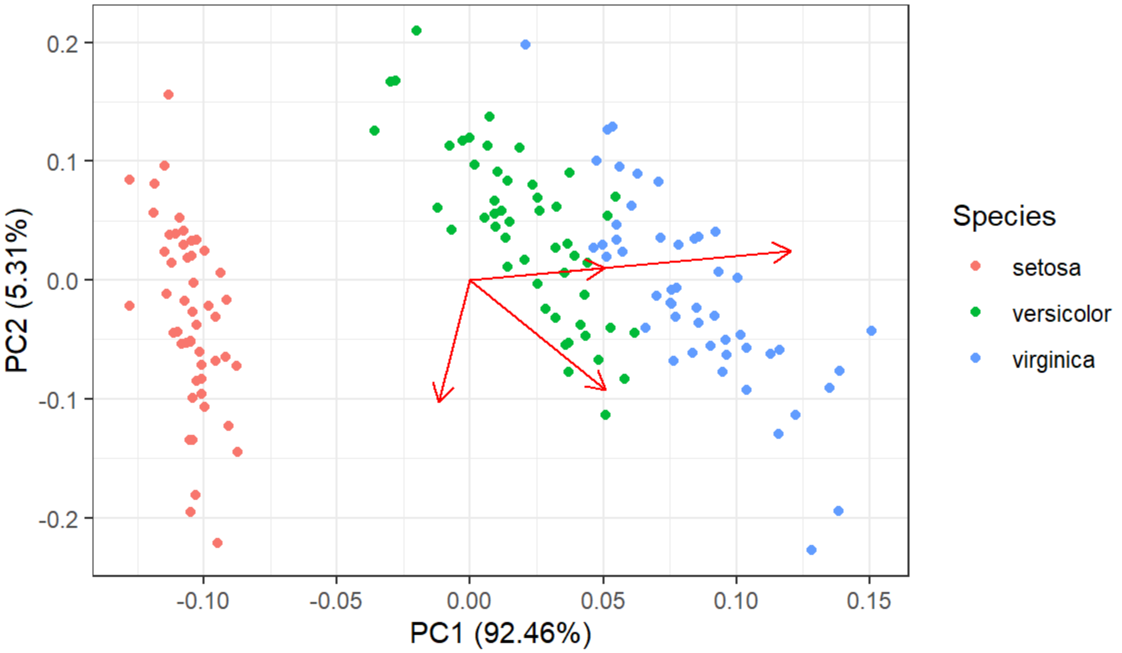
```
df <- iris[,1:4]
autoplot(prcomp(df))
```



About **ggplot2**

Plotting PCA results with `iris` data

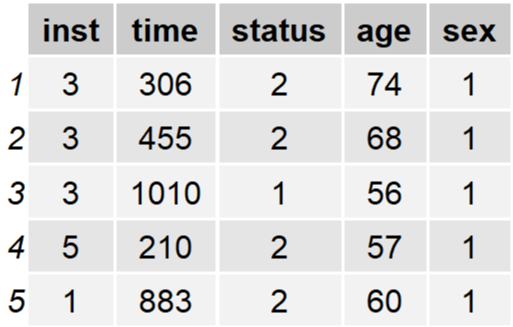
```
autoplot(prcomp(df), data = iris, colour = 'Species', loadings = T) +
  theme_bw()
```



About **ggplot2**

Can also visualize survival curves for the differences in survival times of patients with advanced lung cancer between males and females

```
data("lung")
```

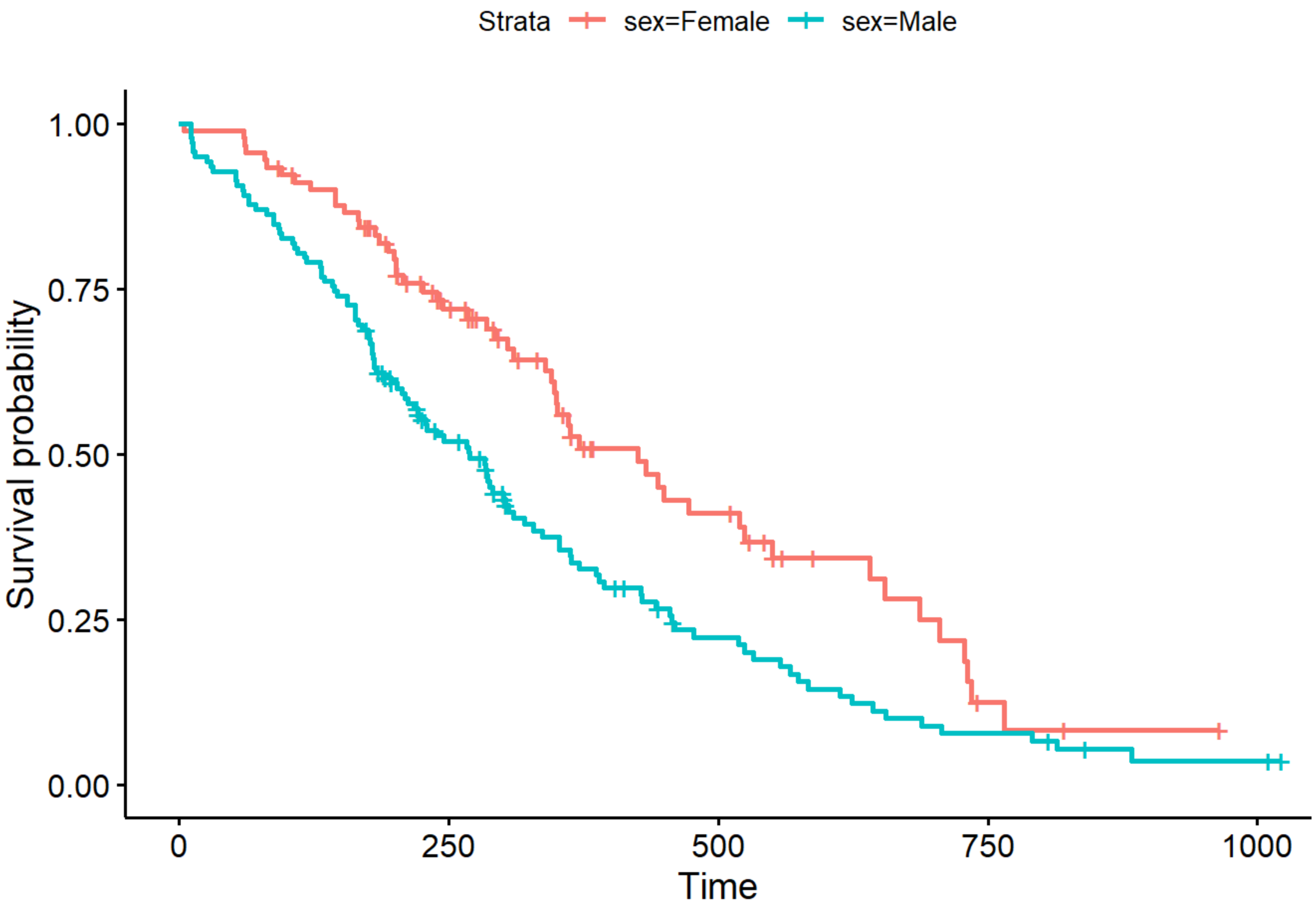


	inst	time	status	age	sex
1	3	306	2	74	1
2	3	455	2	68	1
3	3	1010	1	56	1
4	5	210	2	57	1
5	1	883	2	60	1

About **ggplot2**

```r
lung %>%
  mutate(sex = ifelse(sex == 1, "Male", "Female")) %>%
  survfit(Surv(time, status) ~ sex, data = .) %>%
  ggsurvplot(data = lung)
```

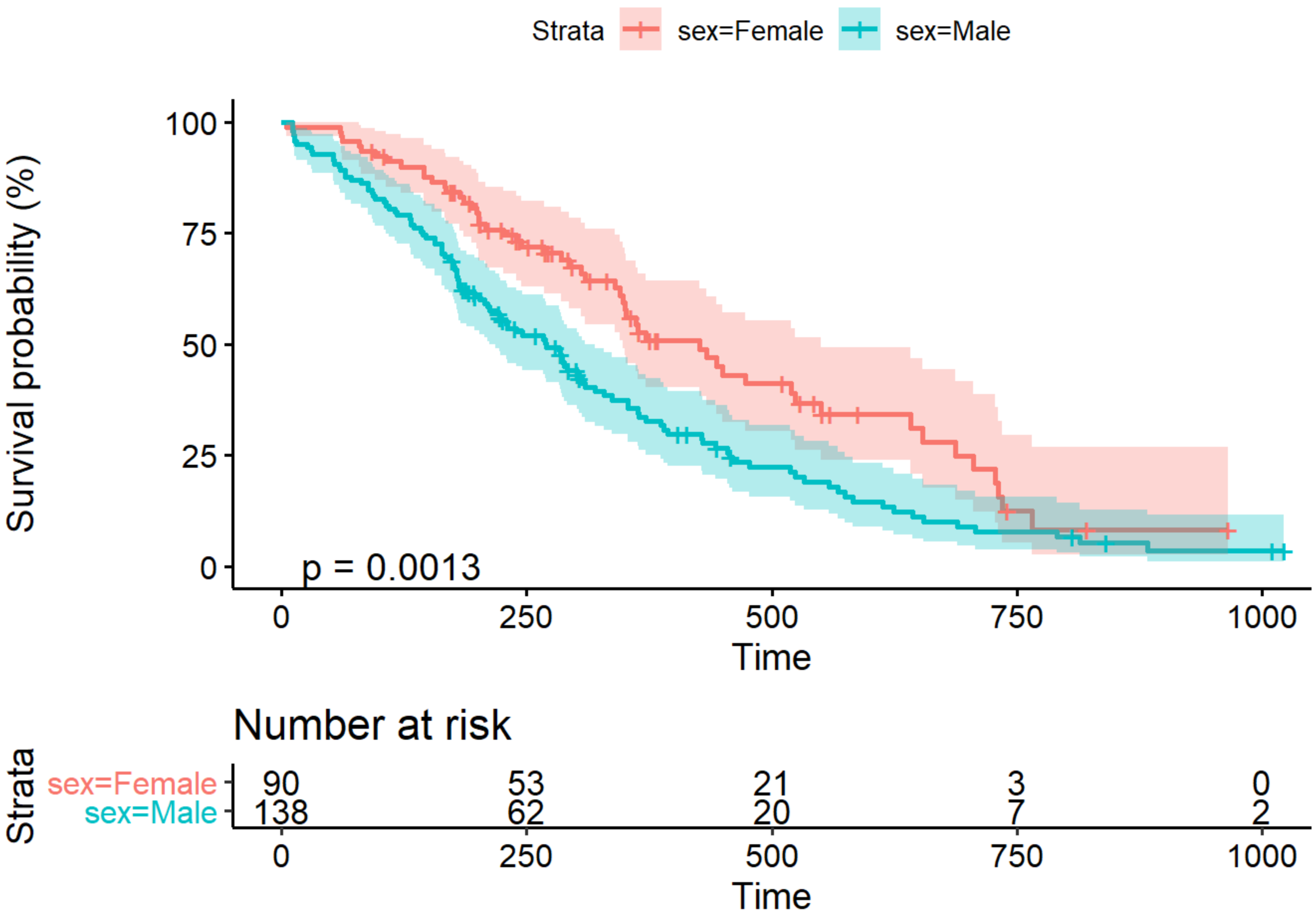
About **`ggplot2`**



About **ggplot2**

```
lung %>%
  mutate(sex = ifelse(sex == 1, "Male", "Female")) %>%
  survfit(Surv(time, status) ~ sex, data = .) %>%
  ggsurvplot(data = lung,
             pval = TRUE, fun = "pct",
             conf.int = T, risk.table = TRUE,
             size = 1)
```

About **ggplot2**


Strata
sex=Female
sex=Male
Survival probability (%)
p = 0.0013
Time
Number at risk
Strata
sex=Female
sex=Male
Time