

Reference sheet

dplyr

- `select()` selecting columns
- `filter()` filter rows
- `group_by()` groups data by categorical levels
- `summarize()` summarize data by functions of choice
- `arrange()` ordering data
- `mutate()` create new columns

dplyr functions

- `n()` (inside `summarize()` only)
- `n_distinct()`

vectors

- `unique()`
- `sort()`
- `length()`
- `1:10`

sapply

- `sapply(v, FUN = f, extra1, extra2, extra3)`

grep

- `grep(pattern, text)`

classes

- `numeric`
- `character`
- `factor`
- `logical`

data.frame

- `my.data[row, column]`

Regression

- `lm(formula, data = my.data)`
- `glm(formula, data = my.data, family = binomial)`
- `predict(fit, newdata = my.data)`
- `predict(fit, newdata = my.data, type = "response")`

- `fit$coefficients`

Conditionals

```
if(<COND1>){
  ...
} else if(<COND2>){
  ...
} else {
  ...
}
```

melt

```
> dat <- data.frame(a = c(4, 5, 5), B = c(10, 20, 30), C = c(100, 200, 300))
>
> melt(dat, "B")
  B variable value
1 10        a     4
2 20        a     5
3 30        a     5
4 10        C    100
5 20        C    200
6 30        C    300
```

Probability

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

```
dt(x, df, ncp, log = FALSE)
pt(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
rt(n, df, ncp)
```

```
dbinom(x, size, prob, log = FALSE)
pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
rbinom(n, size, prob)
```

Some formulas

- T-statistic (one-sample): $\frac{\bar{x} - \mu}{s/\sqrt{n}}$
- T-statistic (two-sample): $\frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$
- $\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma^2}{N})$
- $N(n \times p, n \times p \times (1 - p))$

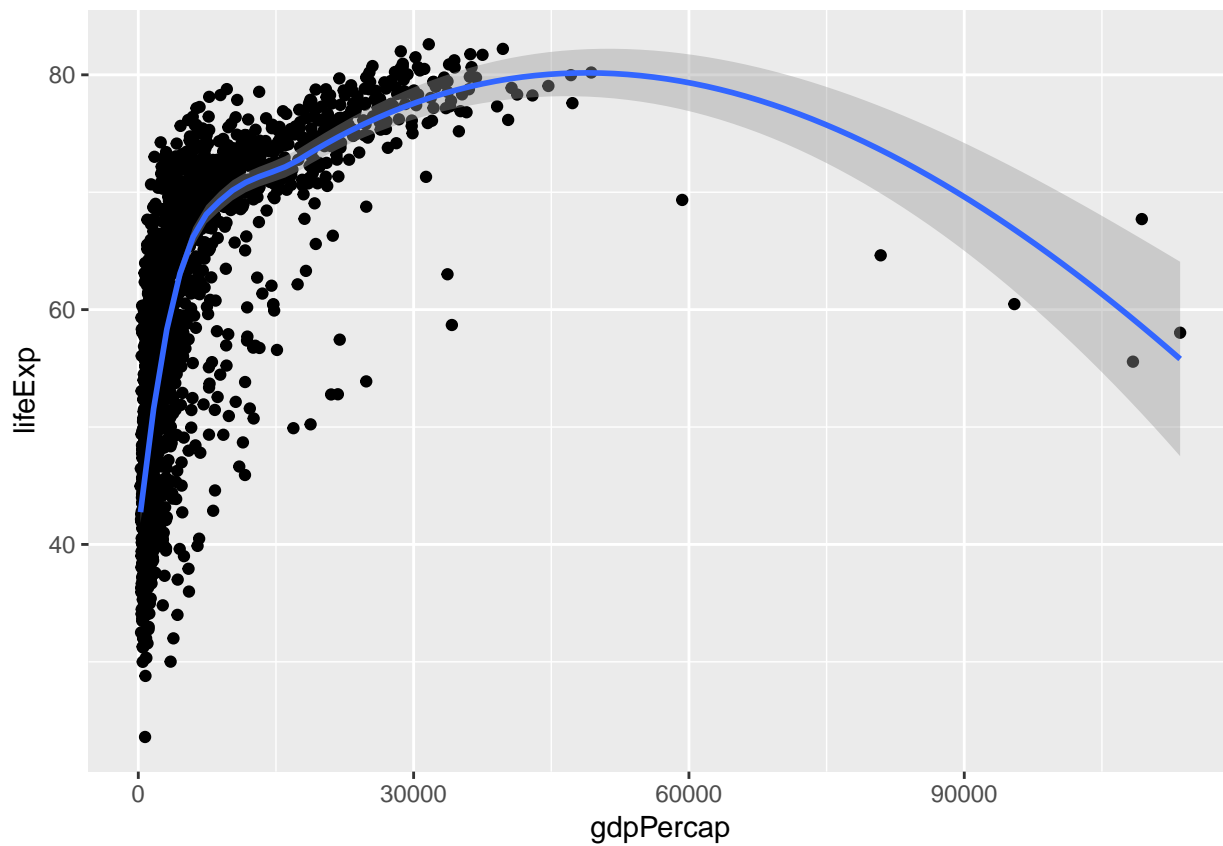
Character manipulation

- `substr(x, start, stop)`
- `str_replace(string, pattern, replacement)`

ggplot

```
p <- ggplot(data = gapminder,  
            mapping = aes(x = gdpPercap,  
                          y = lifeExp))
```

```
p + geom_point() + geom_smooth(method = "loess")
```



```
q <- ggplot(data = gapminder %>% filter(year == 1982),  
            mapping = aes(x = gdpPercap,  
                          y = lifeExp,  
                          color = continent))  
q + geom_smooth(method = "gam", se = F)
```

