# Displaying regression results

# A regression model

We will start with the `mpg` dataset that comes with the **ggplot2** package, and fit a linear regression model to see how city fuel efficiency depends on the year, transmission, number of cylinders, drive and class of vehicle

A lot of the work in this tutorial will use string manipulations using functions from the **stringr** package. This package does have a cheatsheet (https://github.com/rstudio/cheatsheets/raw/master/strings.pdf that you can download.

# Looking at the output

```
library(broom)
out <- tidy(fit, conf.int = TRUE)

out
```

| term<br>&lt;chr&gt; | estimate<br>&lt;dbl&gt; | std.error<br>&lt;dbl&gt; | sta |
|---|---:|---:|---:|
| (Intercept) | 20.56224257 | 1.1227750 | 18.31 |
| year2008 | 0.46240897 | 0.2712424 | 1.70 |
| transmanual | 0.40247464 | 0.3106162 | 1.29 |
| cyl5 | -2.10580025 | 1.0607113 | -1.98 |
| cyl6 | -3.41624564 | 0.3620780 | -9.43 |
| cyl8 | -5.59382007 | 0.4951472 | -11.29 |
| drvf | 2.50309332 | 0.4872506 | 5.13 |
| drvr | -0.08735267 | 0.5793938 | -0.15 |
| classcompact | -1.67746356 | 1.1124558 | -1.50 |
| classmidsize | -2.29142648 | 1.1438440 | -2.00 |

1-10 of 14 rows | 1-4 of 7 colu… Previous **1** 2 Next

We need to fix the `terms` first. The factor variables have the unfortunate default behavior of being represened as with no real formatting. Also we could change the base level of `drv`

# Displaying regression results

to `f` (front wheel drive) from `4` (4-wheel drive), but we'll let it be for now.

# Munging output

We're going to build up the code to get this output data where we'd like it.

First, we need to identify the factor variables so that we can change them. Fortunately, R has already stored this information in the `fit` object.

```
fit$xlevels
```

```
## $year
## [1] "1999" "2008"
## 
## $trans
## [1] "auto"   "manual"
## 
## $cyl
## [1] "4" "5" "6" "8"
## 
## $drv
## [1] "4" "f" "r"
## 
## $class
## [1] "2seater"    "compact"    "midsize"
"minivan"    "pickup"
## [6] "subcompact" "suv"
```

So we can see exactly which variables in the model are categorical and what their levels are.

# Fixing the numbers

We're going to restrict the numbers to 2 decimals

Code  Start Over  Solution  ▶ Run Code

```
for(n in names(fit$xlevels)){
  out <- out %>%
    mutate(term = ifelse(str_detect(term, n),
                         str_replace(term, n, paste(n,'=
                         term))
}

out <- out %>%
  mutate_____(________, ~round(., 2))
```

This is quite reasonable for printing now.

# Displaying regression results

Start Over

```
# for(n in names(fit$xlevels)){
#   out <- out %>%
#     mutate(term = ifelse(str_detect(term,
 n),
#                          str_replace(term,
 n, paste(n,'= ')),
#                          term))
# }
#
# out <- out %>%
#   mutate_if(is.numeric, ~round(., 2))

knitr::kable(out)
```

| term | estimate | std.error | statistic | p.va |
|---|---:|---:|---:|---:|
| (Intercept) | 20.5622426 | 1.1227750 | 18.3137694 | 0.0000 |
| year = 2008 | 0.4624090 | 0.2712424 | 1.7047815 | 0.0896 |
| trans = manual | 0.4024746 | 0.3106162 | 1.2957299 | 0.1964 |
| cyl = 5 | -2.1058003 | 1.0607113 | -1.9852718 | 0.0483 |
| cyl = 6 | -3.4162456 | 0.3620780 | -9.4351106 | 0.0000 |
| cyl = 8 | -5.5938201 | 0.4951472 | -11.2972877 | 0.0000 |
| drv = f | 2.5030933 | 0.4872506 | 5.1371787 | 0.0000 |
| drv = r | -0.0873527 | 0.5793938 | -0.1507656 | 0.8802 |
| class = compact | -1.6774636 | 1.1124558 | -1.5078924 | 0.1330 |
| class = midsize | -2.2914265 | 1.1438440 | -2.0032683 | 0.0463 |
| class = minivan | -4.3516621 | 1.2936355 | -3.3639014 | 0.0009 |
| class = pickup | -3.5335839 | 1.1133642 | -3.1737898 | 0.0017 |
| class = subcompact | -0.5689003 | 1.0690946 | -0.5321328 | 0.5951 |
| class = suv | -3.0282240 | 1.0504330 | -2.8828341 | 0.0043 |

Previous Topic

# Displaying regression results

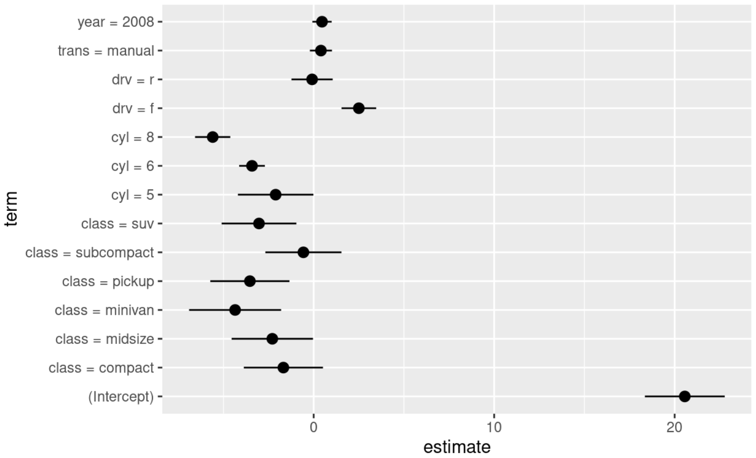
Start Over

# Plotting

We’re now in a position to plot this out

Code Start Over ▶ Run Code

```
# for(n in names(fit$xlevels)){
#   out <- out %>%
#     mutate(term = ifelse(str_detect(term, n),
#                          str_replace(term, n, paste(n,
#                          term))
# }
#
# out <- out %>%
#   mutate_if(is.numeric, ~round(., 2))
#
ggplot(out, aes(x = term, y = estimate,
                ymin = conf.low, ymax = conf.high))+
  geom_pointrange() +
  coord_flip()
```



Modify this code to make the plot prettier. The reference level should be 0

# Displaying regression results

Start Over