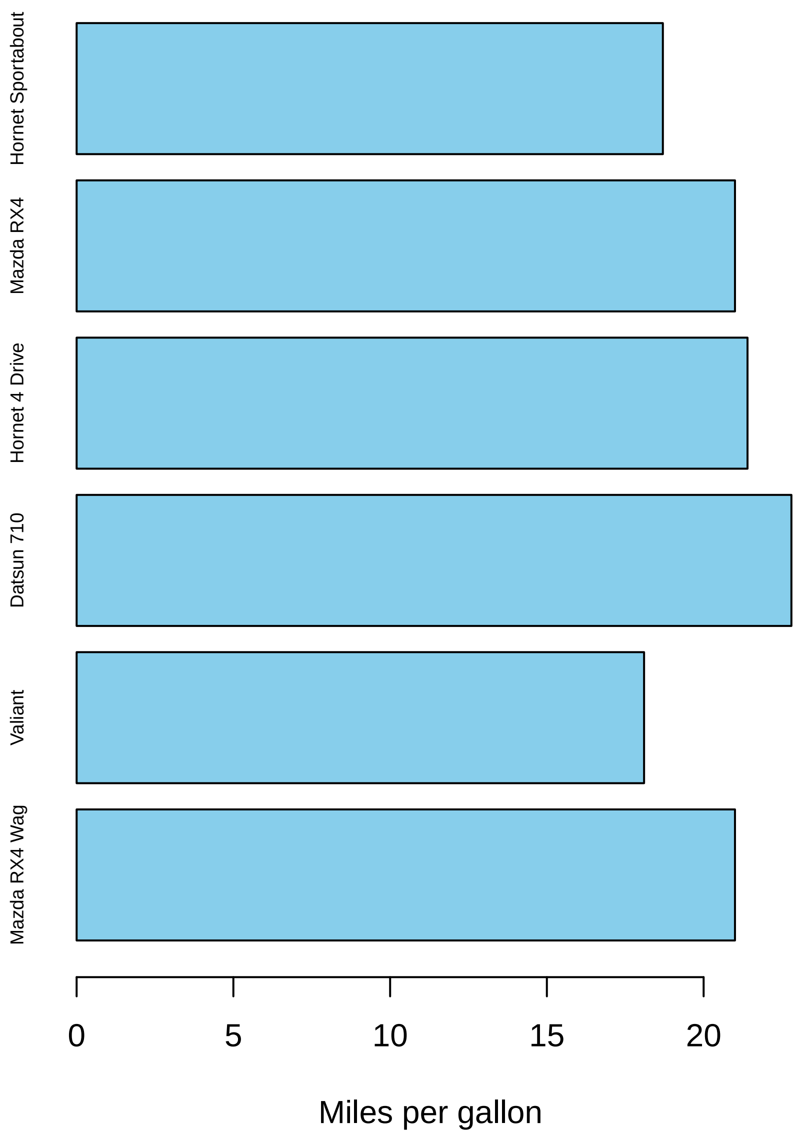
Homework 11

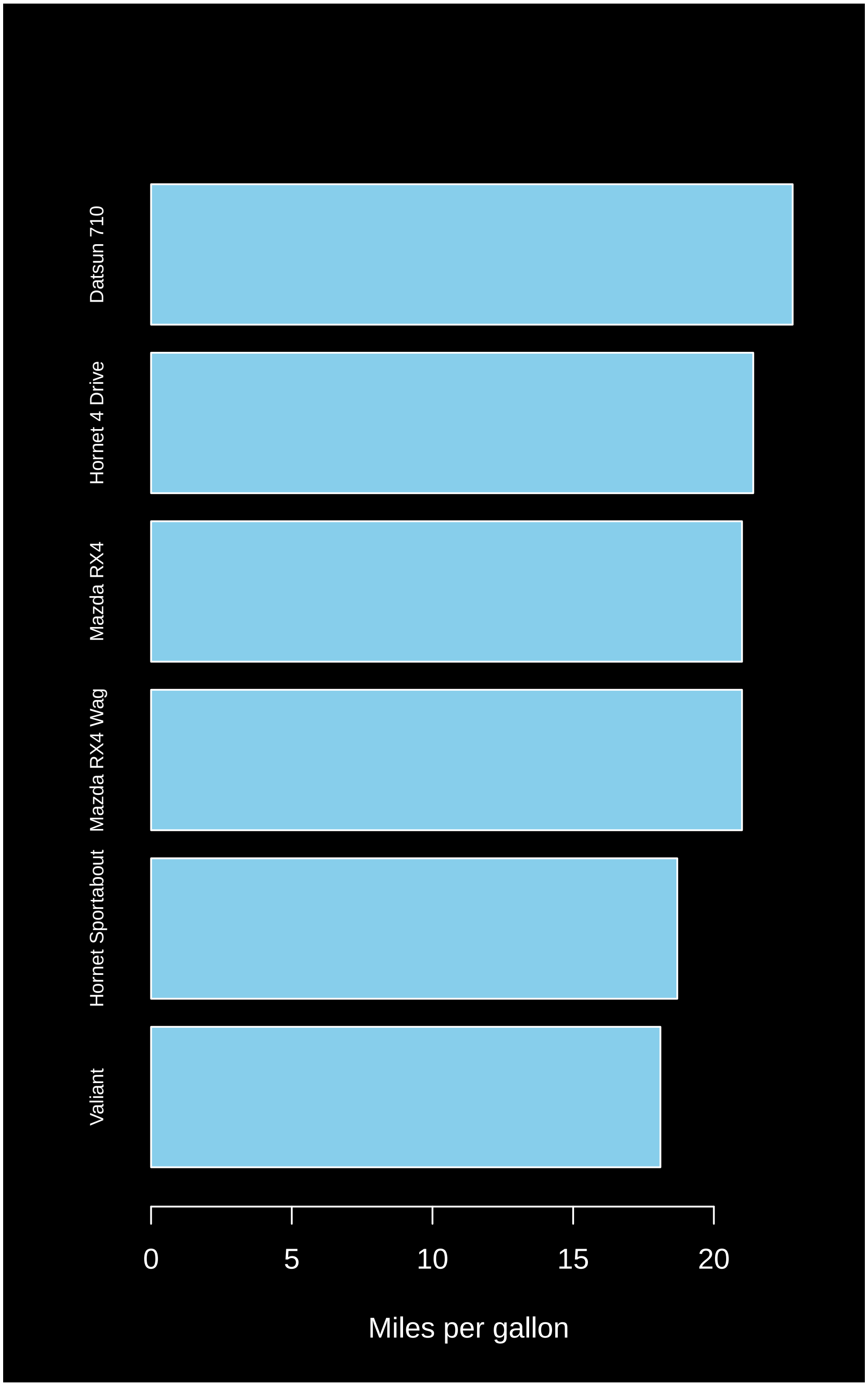
Note: Present both the code and the output for all questions. If you need to change `par()`, remember to save the default setting first and then restore it after finishing your plot. Make the plots with base R graphics only, no `ggplot2` or `lattice` figures for this homework.

1. (1 point) We presented the below barplot on slide 7 of class 11. Reorder the bars by the `mpg` values so that the largest `mpg` will be at the top and the smallest one will be at the bottom.

```
data("mtcars")
set.seed(1)
d = mtcars[sample(6),]
barplot(d$mpg, xlab = "Miles per gallon", col = "skyblue",
        horiz = T, names.arg = row.names(d), cex.names = 0.6)
```



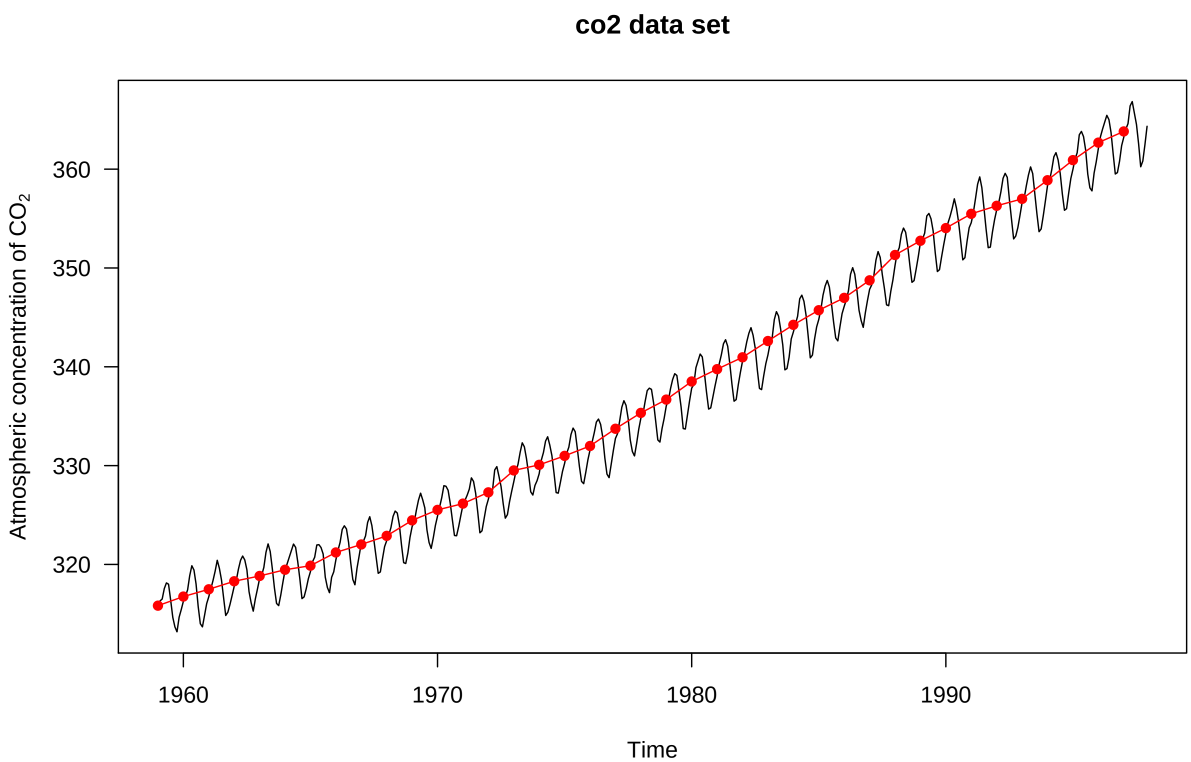
2. (2 point) Change the background of the plot from Question 1 to black and change other elements (e.g. text, axis) to white. Your output should look like the plot below. Present both your code and output. Remember to save the default setting of `par()` and restore it back after the plot.



3. (2 point) The CO_2 concentration in our atmosphere is increasing dramatically over the past hundred years. R has a dataset with a time-series of atmospheric concentration of CO_2 named as `co2`, which you can load via `data("co2")`. In the documentation of `co2` (i.e. `?co2`), you can find an example that will plot the trend of CO_2. After running the example, can you:

   - add the average CO_2 concentration for each year as red solid points onto the plot? (Hint: there are many ways to do this but you may want to check out `?aggregate`.)
   - connect these points with a dashed line.

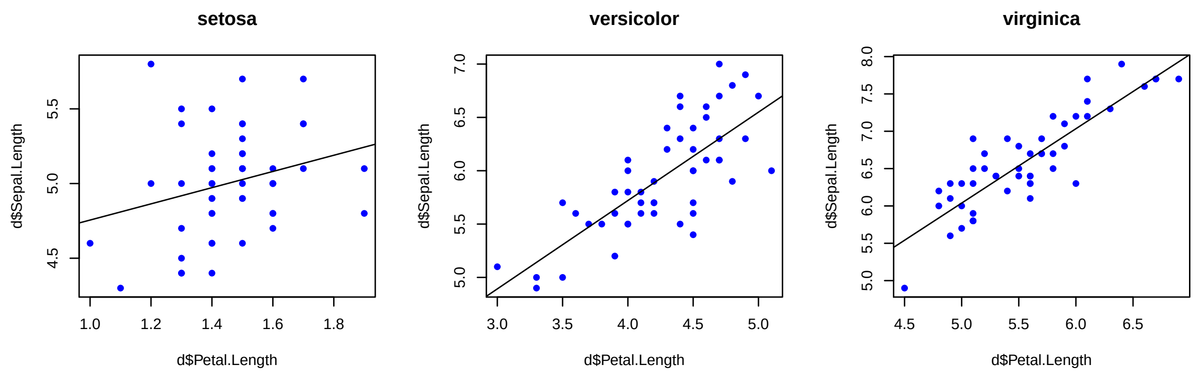
   Your final plot should look like this:



4. (3 point) Let's load the famous Iris dataset that comes with R via `data('iris')`. By reading its documentation we know that it has flower size measurements for three species. Use a `for` loop to create multiple panels (one for each species) within a single figure (hint: change `par()` to allow multiple sub-figures). In each panel:

   - display `Petal.Length` against `Sepal.Length` with solid blue points
   - perform a linear regression between `Petal.Length` and `Sepal.Length`. The linear regression model can be specified as `mod_name <- lm(Sepal.Length ~ Petal.Length, data= SUBDATA_FOR_SPECIES)`
   - add a line based on the linear regression results (hint: `?abline`)
   - add the species name as the title for each panel.

   Your final plot should look like this:



5. (2 point) Visualization of data is extremely important. In most cases, just looking at some summarized statistics of our data can be misleading. One famous example is the Anscombe's quartet, which has four pairs of variables. Here is the dataset:

```
anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

We can see that all four x variables have the same average value and all four y variables have the same average value.

```
colMeans(anscombe)
```

```
##  x1  x2  x3  x4  y1  y2  y3  y4
## 9.0 9.0 9.0 9.0 7.5 7.5 7.5 7.5
```

In addition, if we look at the linear regression tables, we can see that they are almost identical.

```
coef(summary(lm(y1 ~ x1, data = anscombe)))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.0      1.125    2.67  0.02573
## x1               0.5      0.118    4.24  0.00217
```

```
coef(summary(lm(y2 ~ x2, data = anscombe)))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.0      1.125    2.67  0.02576
## x2               0.5      0.118    4.24  0.00218
```

```
coef(summary(lm(y3 ~ x3, data = anscombe)))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.0      1.124    2.67  0.02562
## x3               0.5      0.118    4.24  0.00218
```

```r
coef(summary(lm(y4 ~ x4, data = anscombe)))
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.0      1.124    2.67  0.02559
## x4               0.5      0.118    4.24  0.00216
```

Similar to question 4, use a `for` loop to plot these four pairs of variables as separate panels within one figure (2 rows by 2 columns). Each panel is a scatter plot of `x` and `y`, with solid blue points and a line based on the linear regression model. Does the output convince you of the importance of data visualization?