Week3DA

LouisY

1 Introduction

1.1 Sub1

Table **??** displays the first 5 rows of the iris data…

1.2 Sub2

1.3 R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

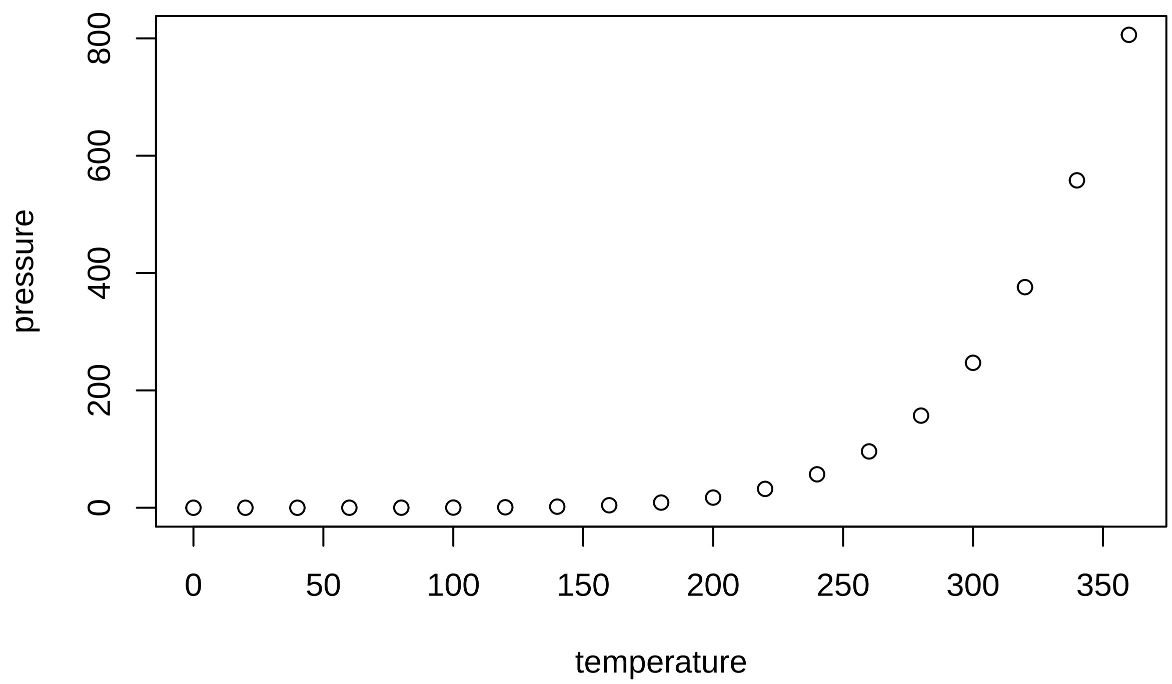
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

1.4 Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
iris %>%
  group_by(Species) %>%
  summarise(n=n(),Mean=round(mean(Sepal.Width),digits=1),
    St.Dev=round(sd(Sepal.Width),digits=1), Min=min(Sepal.Width),
    Q1 = quantile(Sepal.Width,0.25), Median=median(Sepal.Width),
    Q3 = quantile(Sepal.Width,0.75), Max=max(Sepal.Width)) %>%
  kable(caption = '\\label{tab:summaries} Summary statistics on the sepal
    width by species of irises.') %>%
  kable_styling(font_size = 10, latex_options = "hold_position")
```

Table 1: Summary statistics on the sepal width by species of irises.

Species	n	Mean	St.Dev	Min	Q1	Median	Q3	Max
setosa	50	3.4	0.4	2.3	3.200	3.4	3.675	4.4
versicolor	50	2.8	0.3	2.0	2.525	2.8	3.000	3.4
virginica	50	3.0	0.3	2.2	2.800	3.0	3.175	3.8

`summarise()` ungrouping output (override with `.groups` argument)

```
my_skim <- skim_with(base = sfl(n = length))
iris %>%
  group_by(Species) %>%
  select(Sepal.Length,Species) %>%
  my_skim() %>%
  transmute(Variable=skim_variable, Species=Species, n=n, Mean=numeric.mean, SD=numeric.sd,
            Min=numeric.p0, Median=numeric.p50,  Max=numeric.p100,
            IQR = numeric.p75-numeric.p50) %>%
   kable(caption = ’\\label{tab:summaryskim} Summary statistics of the sepal length by species of irise
         booktabs = TRUE, linesep = "", digits = 2) %>%
   kable_styling(font_size = 10, latex_options = "hold_position")
```

Table 2: Summary statistics of the sepal length by species of irises (produced using skim() function).

Variable	Species	n	Mean	SD	Min	Median	Max	IQR
Sepal.Length	setosa	50	5.01	0.35	4.3	5.0	5.8	0.2
Sepal.Length	versicolor	50	5.94	0.52	4.9	5.9	7.0	0.4
Sepal.Length	virginica	50	6.59	0.64	4.9	6.5	7.9	0.4

Tables of model estimates Often we also need to report the results of fitting a model to our data. For instance if we modeled the sepal length on the different species in the iris data by:

```
model <- lm(Sepal.Length ~ Species, data = iris)
get_regression_table(model) %>%
  dplyr::select(term,estimate) %>%
  #Note that it seems necessary to include "dplyr::" here!!
  kable(caption = ’\\label{tab:reg} Estimates of the parameters from the fitted linear regression model
  kable_styling(latex_options = ’HOLD_position’)
```

Table 3: Estimates of the parameters from the fitted linear regression model.

term	estimate
intercept	5.006
Speciesversicolor	0.930
Speciesvirginica	1.582

Tables ‘by hand’ Tables can also be produced “by hand”" in Markdown. For example, the table above corresponding to the first 5 rows of the iris data can be produced by hand by typing the following text (without any other text) into a .Rmd file:

Table 4: The fist 5 rows of the iris data.

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

```
{r scatplot, echo = FALSE, eval = TRUE, out.width = '70%',  fig.align = "center", fig.pos
= "h", warning = FALSE,  fig.cap = "\\label{fig:scat} Relationship between teaching
```

and beauty scores. The best-fitting line has been superimposed."} ggplot(evals.scores, aes(x = bty_avg, y = score)) +   geom_point() +   labs(x = "Beauty Score", y = "Teaching Score") +   geom_smooth(method = "lm", se = FALSE) geom_smooth() using formula ‘y ~ x’