dviz-practice

Practice for 4_plot.org

1. README

- Practice for `plot` customization as discussed in the lecture
- Includes example of downloading CSV data into a data frame
- Create R code blocks to complete this file
- Submit Org-file on Canvas when you're done

2. TODO Identify yourself

1. Add your name and `(pledged)` at the top
2. Run the `#+PROPERTY` and `#+STARTUP` lines with `C-c C-c`

3. TODO Getting the data

1. Download `data.csv` to the same directory as the practice file: tinyurl.com/spdnvxbr
2. Save data from the local CSV file to an R data frame `df`:

```
df <- read.csv(
  file="./data/dataset.csv",
  header=TRUE)
str(df)
```

```
'data.frame':	10 obs. of  4 variables:
$ Weight: int  55 85 75 42 93 63 58 75 89 67
$ Height: int  161 185 174 154 188 178 170 167 181 178
$ Sex   : chr  "female" "male" "male" "female" ...
$ Name  : chr  "Jane" "Jim" "Joe" "Carla" ...
```

Alternatively, use the URL for the `file` parameter:

```
df1 <- read.csv(
  file="https://tinyurl.com/spdnvxbr",
  header=TRUE)
str(df1)
```

```
'data.frame':	10 obs. of  4 variables:
$ Weight: int  55 85 75 42 93 63 58 75 89 67
$ Height: int  161 185 174 154 188 178 170 167 181 178
$ Sex   : chr  "female" "male" "male" "female" ...
$ Name  : chr  "Jane" "Jim" "Joe" "Carla" ...
```

4. TODO Plotting the data

1. For the R code blocks with graphics output, you need these header arguments in this format exactly when writing to a file `plot.png` in the current directory:

   ```
   :results graphics file :file plot.png
   ```

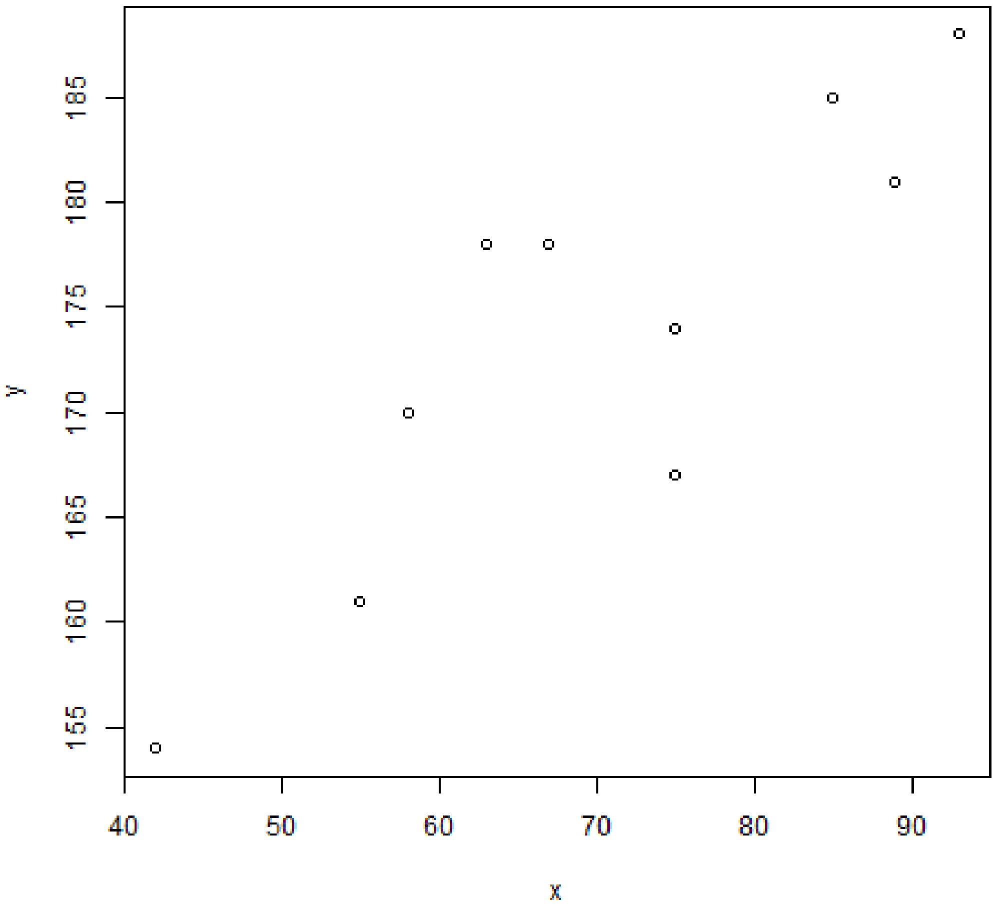
2. Save the `Weight` vector as `x` and the `Height` vector as `y` for later.

```
x <- df$Weight
x
y <- df$Height
y
```

```
[1]  55  85  75  42  93  63  58  75  89  67
[1] 161 185 174 154 188 178 170 167 181 178
```

3. Create a scatterplot of the weight on the `x` axis and the height on the `y` axis. Save the plot as `scatter.png`.

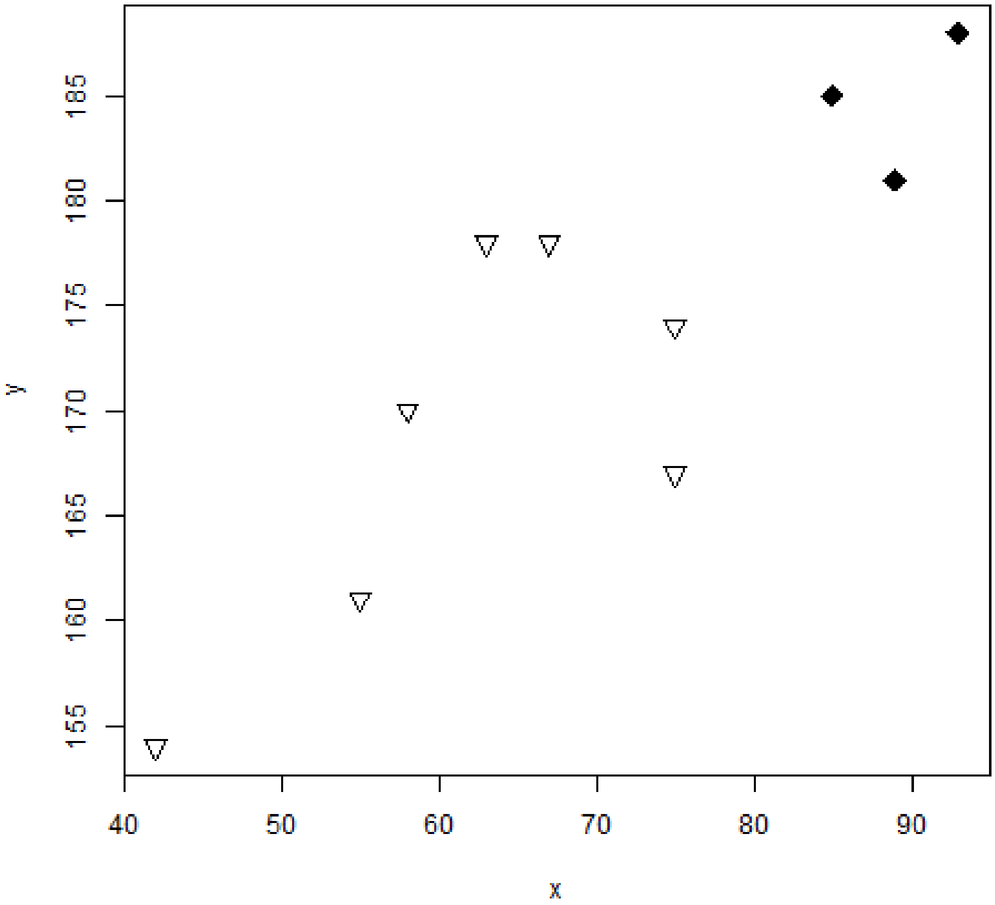
```
plot(y ~ x)
```



5. TODO Customize the plot

1. Customize the plot by using different point characters to identify all people, who are equal to or taller than 180 cm, and save the resulting plot as `pch.png`.
   - Define an appropriate `index` vector using the `which` function
   - Call `plot` as before but don't draw any points
   - Call `points[index]` for outliers ,and `points[-index]` for others
   - Inside the `points` function calls, use `pch=18` and `cex=1` for the outliers, and `pch=6`, `cex=1.3` for the other points

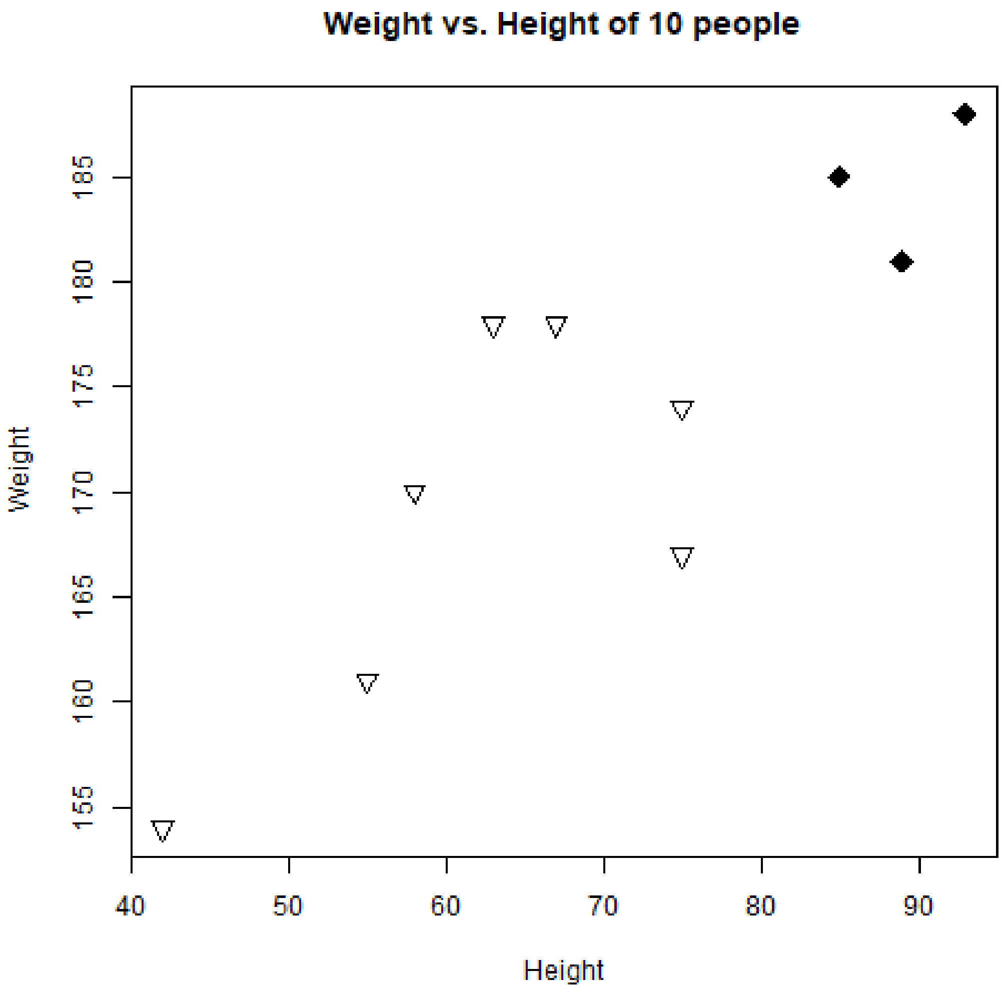
```
plot(y ~ x, type="n")
index <- which(y > 180)
points(x[index], y[index], pch=18, cex=2)
points(x[-index], y[-index], pch=6, cex=1.3)
```



2. Label the axes and give the plot a suitable title. Save your final plot in `final.png`.

```
plot(y ~ x, type="n",
     main="Weight vs. Height of 10 people",
     ylab="Weight", xlab="Height")
```

```
index <- which(y > 180)
points(x[index], y[index], pch=18, cex=2)
points(x[-index], y[-index], pch=6, cex=1.3)
```



6. TODO Add a trendline

In the last plot, we indexed all male participants taller than or equal than 180 cm (about 6'). Now we want to 1) add a trendline, 2) add a suitable legend.

1. To add a trendline, define a model using the `lm` function on the two coordinate vectors `y` and `x`, and store it in `trend`

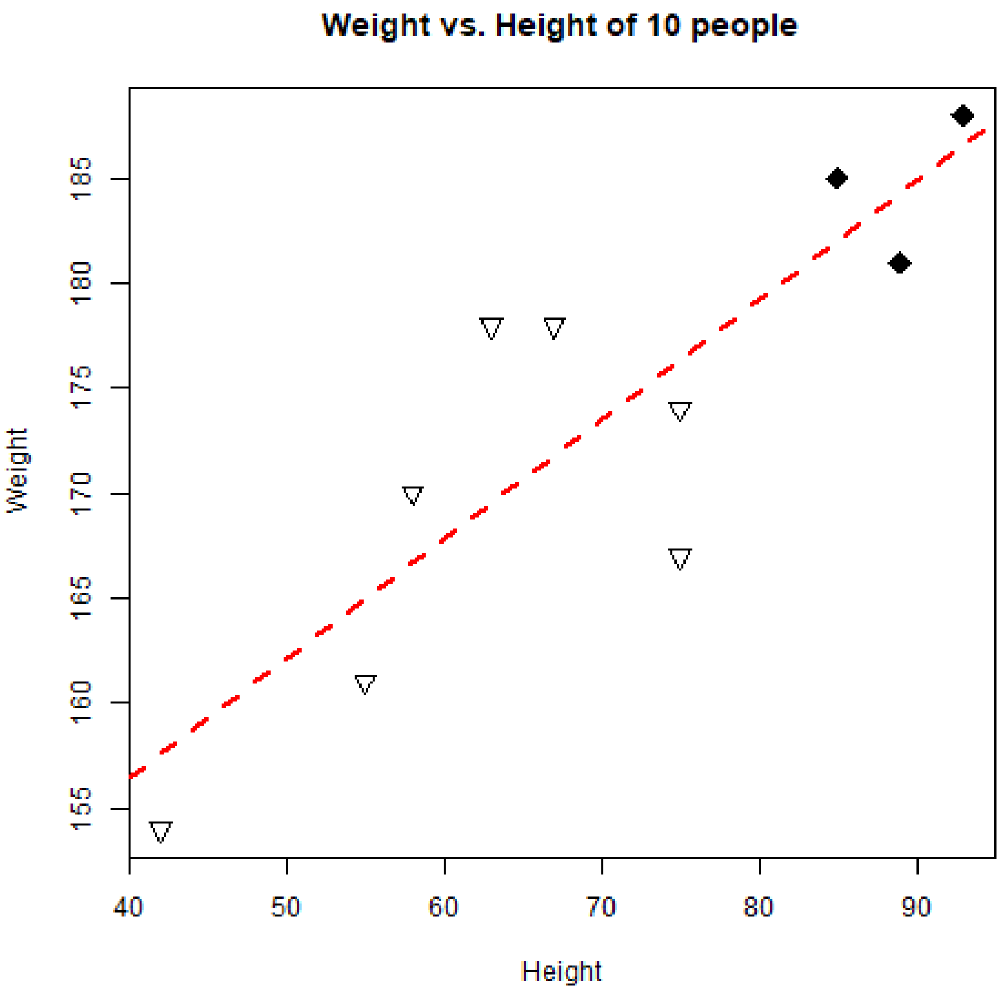
```
trend <- lm(y ~ x)
trend
```

```

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x

  133.7424       0.5678
```

2. Use the model's intercept and slope values to draw a red, dashed, thick trend line. Store the resulting graph in `trend.png`.

   You need to recreated the calls to `plot` and `points` for this graph. Draw this graph first, then add the trendline.

```
plot(y ~ x, type="n",
     main="Weight vs. Height of 10 people",
     ylab="Weight", xlab="Height")
points(x[index], y[index], pch=18, cex=2)
points(x[-index], y[-index], pch=6, cex=1.3)
abline(trend, lty = 2, lwd = 2, col = "red")
```

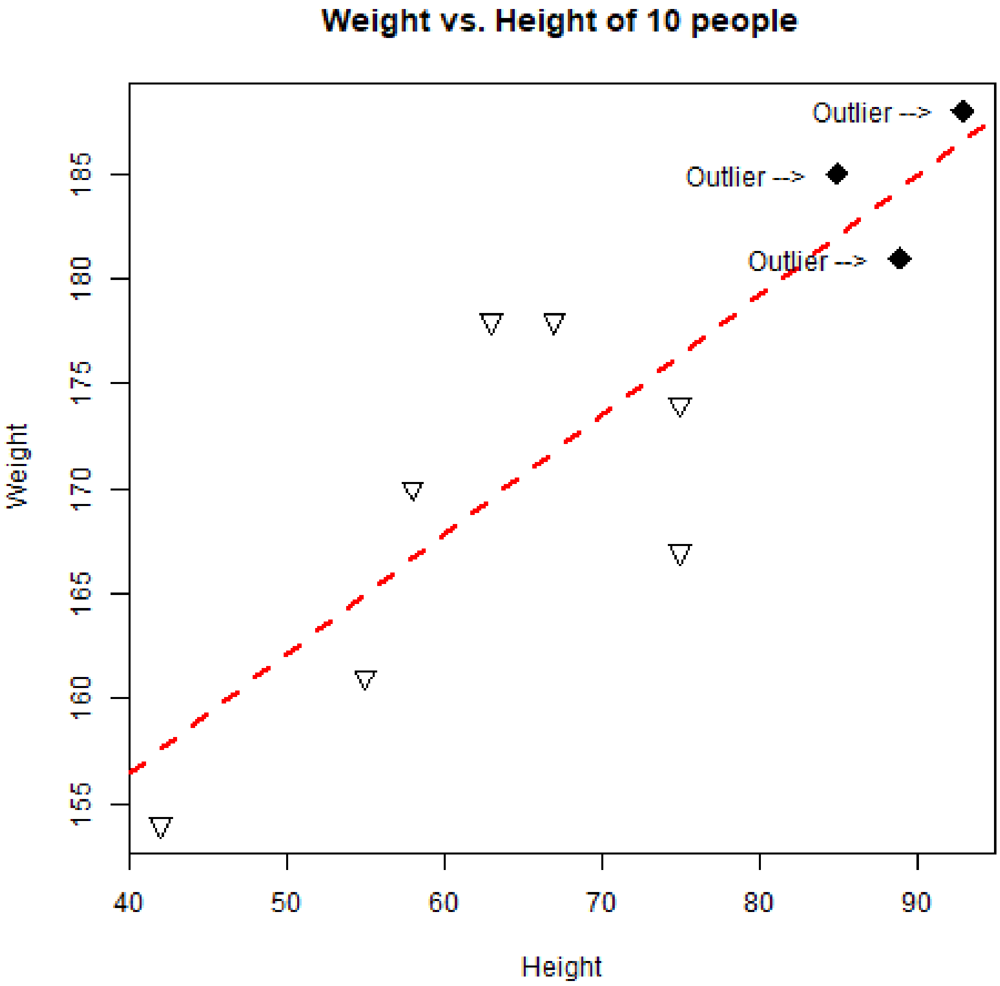


7. TODO Add text for outliers

Add the text "Outlier" to the three outlying points of the plot. You can use the following label: `paste("Outlier -->")`. Left-align the text and make sure that the text does not obscure the data.

Use the previous code block and store the graph in `text.png`.

```
plot(y ~ x, type="n",
     main="Weight vs. Height of 10 people",
     ylab="Weight", xlab="Height")
points(x[index], y[index], pch=18, cex=2)
points(x[-index], y[-index], pch=6, cex=1.3)
abline(trend, lty = 2, lwd = 2, col = "red")
outliers <- paste("Outlier -->")
text(
  x=x[index]-2,
  y=y[index],
  labels=outliers,
  adj=1)
```

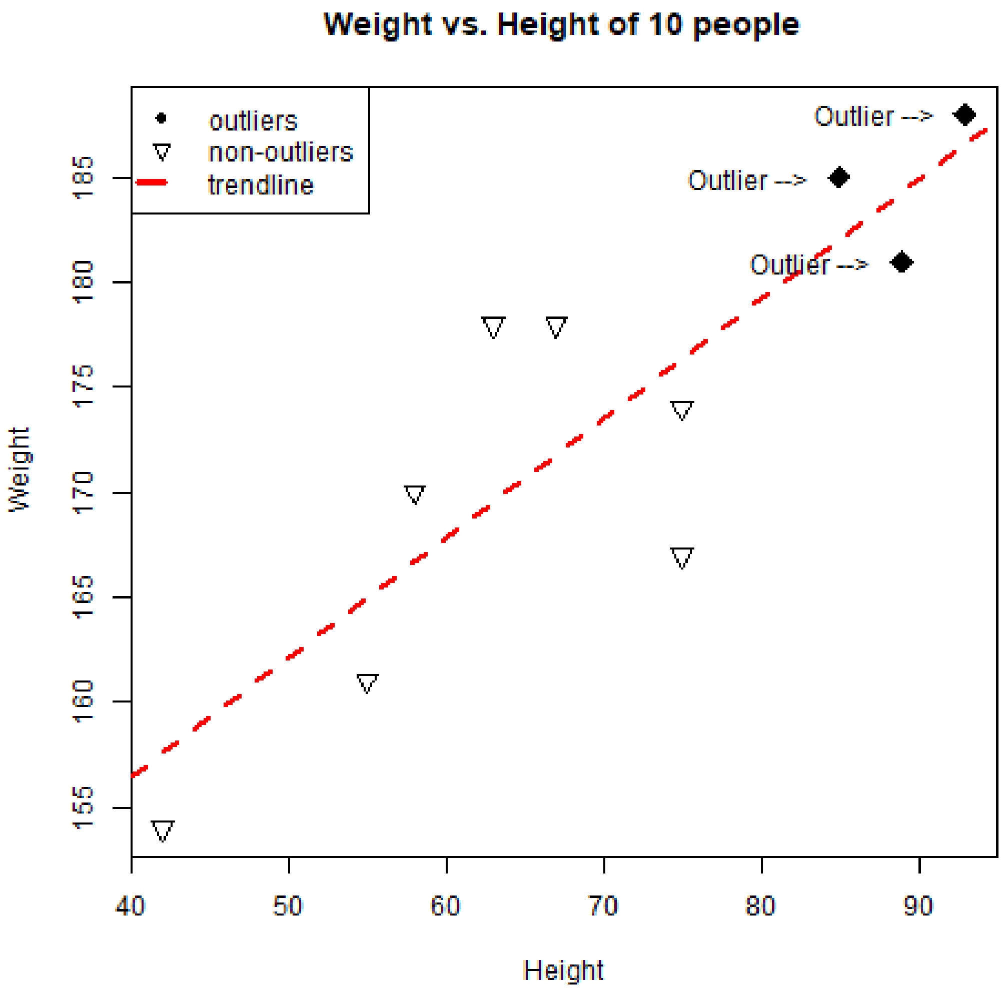


8. TODO Add a legend

Add a legend for the trendline and for the two plot point types. Store the plot in the file `people.png`.

Tip: specify vectors for `pch`, `lty` and `lwd` and specify irrelevant values (e.g. point size for the trendline, or line type for the points) as `NA`.

```
plot(y ~ x, type="n",
     main="Weight vs. Height of 10 people",
     ylab="Weight", xlab="Height")
points(x[index], y[index], pch=18, cex=2)
points(x[-index], y[-index], pch=6, cex=1.3)
abline(trend, lty = 2, lwd = 2, col = "red")
outliers <- paste("Outlier -->")
text(
  x=x[index]-2,
  y=y[index],
  labels=outliers,
  adj=1)

legend(
x = "topleft",
legend=c("outliers","non-outliers","trendline"),
pch=c(18,6,NA),
lty=c(NA,NA,2),
lwd=c(NA,NA,3),
col=c("black","black","red"))
```


Weight vs. Height of 10 people
● outliers
▽ non-outliers
trendline
Outlier -->
Outlier -->
Outlier -->
Weight
Height