Assignment 9

Instructions: Make an R Notebook and in it answer the question or questions below. When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook, probably an `html` or `pdf` file. An `html` file is easier for the grader to deal with. Do *not* hand in the Notebook itself. You want to show that you can (i) write code that will answer the questions, (ii) run that code and get some sensible output, (iii) write some words that show you know what is going on and that reflect your conclusions about the data. Your goal is to convince the grader that you *understand* what you are doing: not only doing the right thing, but making it clear that you know *why* it's the right thing.

Do *not* expect to get help on this assignment. The purpose of the assignments is for you to see how much *you* have understood. You will find that you also learn something from grappling with the assignments. The time to get help is after you watch the lectures and work through the problems from PASIAS, via tutorial and the discussion board, that is *before* you start work on the assignment. The only reasons to contact the instructor while working on an assignment are to report (i) something missing like a data file that cannot possibly be read, (ii) something *beyond your control* that makes it impossible to finish the assignment in time after you have started it.

There is a time limit on this assignment (you will see Quercus counting down the time remaining).

1. The decathlon is a men's track-and-field competition in which competitors complete 10 events over two days as follows, requiring the skills shown:

Event	Skills
100m	Running, speed
Long jump	Jumping, speed
Shot put	Throwing, strength
High jump	Jumping, agility
400m	Running, speed
110m hurdles	Running, jumping, speed
Discus	Throwing, strength (and maybe agility)
Pole vault	Jumping, agility
Javelin	Throwing, agility
1500m	Running, endurance

   These are a mixture of running, jumping and throwing disciplines. The performance (time, distance or height) achieved in each event is converted to a number of points using standard tables, and the winner of the entire decathlon is the competitor with the largest total of points. The basic idea is that a "good" performance in an event is worth 1000 points, and the score decreases if the athlete takes more seconds (running) or achieves fewer metres (jumping/throwing). A good decathlete has to be at least reasonably good at all the disciplines.

   For the decathlon competition at the 2013 Track and Field World Championship, a record was kept of each competitor's performance in each event (for the competitors that competed in all ten events). These values are in http://www.utsc.utoronto.ca/~butler/d29/dec2013.txt, separated by single spaces. These are the actual performances, in seconds for running events and in metres for jumping and throwing events. The columns containing the running events have an `x` on the front of their names, since column names cannot start with a number (as `100m` would, so its name is `x100m`).

(a) Read in and display (some of) the data.

Solution:

This one is `read_delim`:

```
my_url <- "http://www.utsc.utoronto.ca/~butler/d29/dec2013.txt"
decathlon <- read_delim(my_url," ")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   name = col_character(),
##   x100m = col_double(),
##   long.jump = col_double(),
##   shot.put = col_double(),
##   high.jump = col_double(),
##   x400m = col_double(),
##   x110mh = col_double(),
##   discus = col_double(),
##   pole.vault = col_double(),
##   javelin = col_double(),
##   x1500m = col_double()
## )
```

```
decathlon
```

```
## # A tibble: 24 x 11
##    name        x100m long.jump shot.put high.jump x400m x110mh discus pole.vault
##    <chr>       <dbl>     <dbl>    <dbl>     <dbl> <dbl>  <dbl>  <dbl>      <dbl>
##  1 Ashton Eat~  10.4      7.73     14.4      1.93  46.0   13.7   45          5.2
##  2 Damian War~  10.4      7.39     14.2      2.05  48.4   14.0   44.1        4.8
##  3 Rico Freim~  10.6      7.22     14.8      1.99  48.0   13.9   48.7        4.9
##  4 Mihail Dud~  10.7      7.51     13.4      1.96  47.7   14.6   44.1        4.9
##  5 Michael Sc~  10.7      7.85     14.6      1.99  47.7   14.3   46.4        5
##  6 Carlos Chi~  10.8      7.54     14.5      1.96  48.8   14.0   45.8        5.1
##  7 Gunnar Nix~  10.8      7.8      14.7      2.14  48.6   14.6   42.4        4.6
##  8 Eelco Sint~  10.8      7.65     14.1      2.02  48.2   14.2   39.2        5.3
##  9 Pascal Beh~  11.0      7.19     15.9      1.99  48.4   14.5   45.7        4.7
## 10 Willem Coe~  11.0      7.44     13.9      2.05  48.3   14.3   43.2        4.5
## # ... with 14 more rows, and 2 more variables: javelin <dbl>, x1500m <dbl>
```

Extras:

1. You might be wondering how those spaces in the athletes' names made it past `read_delim`, which goes to the next column when it encounters a space. The answer (you can look in the data file to verify) is that the names had quotes around them. What `read_delim` *actually* does is to separate data values by spaces *provided* that those spaces are not inside quotes. `read_csv` does the same thing with commas that are inside quotes, for example if you have a name like "Song, Eric", with the last name first, in a spreadsheet. When you save this as a `csv`, your spreadsheet software will put quotes around the name, so that when your `csv` is read into Excel or R, the name will come out properly.
2. So what happens if a column name starts with a number? Let's try it:

```
data_text <- "
  person Name, 1, 2
  fred, 10, 11
  ginger, 12, 8
"
d <- read_csv(data_text)
d
```

```
## # A tibble: 2 x 3
##   `person Name`   `1`   `2`
##   <chr>         <dbl> <dbl>
## 1 fred             10    11
## 2 ginger           12     8
```

So far so good. But what if you want to `select` one of those columns whose name is a number?

```
d %>% select(1)
```

```
## # A tibble: 2 x 1
##   `person Name`
##   <chr>
## 1 fred
## 2 ginger
```

This selects the column *numbered* 1, that is, the first column, with the names in it. To select the column *called* `1`, you have to do this:

```
d %>% select(`1`)
```

```
## # A tibble: 2 x 1
##     `1`
##   <dbl>
## 1    10
## 2    12
```

Any column name that starts with a number, or which contains illegal characters like spaces, has to be referred to this way, with "backticks" around it. This is the unshifted version of the key with "squiggle" on it, to the left of your 1 key, not the apostrophe that is near your enter key. See problem 5.1 in PASIAS for more on this.

There is another way around this, which is to tidy up your column names first, so that you don't have to deal with them. The easy way to do this is to use `clean_names` from the `janitor` package:

```
library(janitor)
d %>% clean_names()
```

```
## # A tibble: 2 x 3
##   person_name    x1    x2
##   <chr>       <dbl> <dbl>
## 1 fred           10    11
## 2 ginger         12     8
```

Among other things, spaces in column names are replaced with underscores, and column names starting with numbers get an `x` on the front. Also, column names are made all-lowercase, so

that you're not struggling to remember whether columns have uppercase or lowercase letters in their names.

(b) Run a principal components analysis on all the appropriate columns, and display the results.

Solution:

The columns in a principal components analysis have to be quantitative, so get rid of the names first:

```
decathlon %>% select(-name) %>%
  princomp(cor = TRUE) -> decathlon.1
summary(decathlon.1)
```

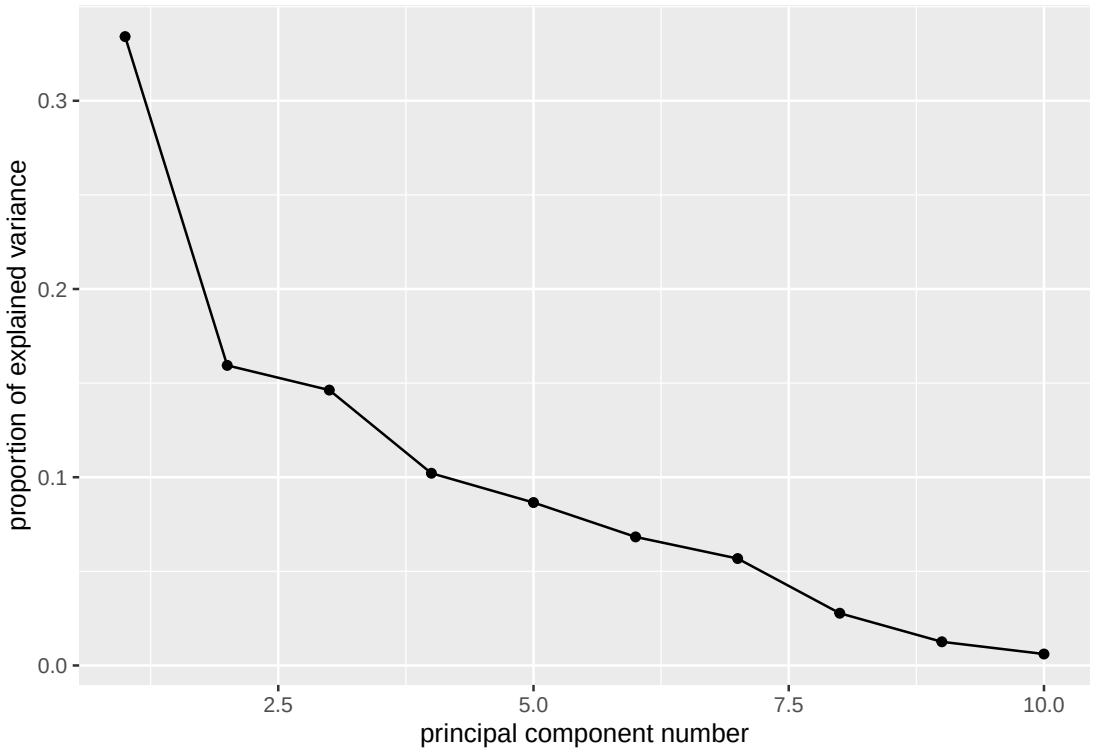
```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## Standard deviation     1.8278139 1.2626364 1.2095405 1.0105352 0.93037481
## Proportion of Variance 0.3340903 0.1594251 0.1462988 0.1021181 0.08655973
## Cumulative Proportion  0.3340903 0.4935154 0.6398143 0.7419324 0.82849212
##                            Comp.6     Comp.7     Comp.8     Comp.9     Comp.10
## Standard deviation     0.82639794 0.75365771 0.52674982 0.35464020 0.246799947
## Proportion of Variance 0.06829336 0.05679999 0.02774654 0.01257697 0.006091021
## Cumulative Proportion  0.89678548 0.95358547 0.98133201 0.99390898 1.000000000
```

(c) Make a scree plot for these data.

Solution:

The easiest way is to run `ggscreeplot` (from package `ggbiplot`) on the fitted model object:

```
ggscreeplot(decathlon.1)
```



(d) Give brief reasons both for and against using three components.

Solution:

The usual thing to look for is an elbow. There is one at 4, which would support using $4-1=3$ components. Against that, I think the best thing is to say that this is still a long way up the mountain, or, said another way, three components doesn't explain much of the variability. From the previous output, they explain only 64% of the variability, and it would be nice to explain more.

The bigger elbow at 2 (suggesting one component) is not really relevant here, because that is even further up the mountain, explaining even less of the variability. Going the other way, the little elbow at 9 (suggesting eight components) is not helpful either, because we are trying to explain ten variables by something a lot less than ten, which eight is not.

Extra: perhaps we shouldn't have expected great things from this analysis, because we might have expected that the ten events in the decathlon are there for a reason, that they measure different aspects of athletic ability, and that there isn't really a small number of "components" or "factors" describing how the athletes perform. Problem 26.7 in PASIAS does a cluster analysis on this same data set, and addresses some of these issues again.

(e) How do the first three components seem to depend on the original variables? What, therefore, do the events in each of these components seem to have in common?

Solution:

This is an invitation to look at the component loadings:

```
decathlon.1$loadings
```

```
##
## Loadings:
##            Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## x100m       0.457                       0.328  0.459  0.150  0.137  0.352
## long.jump  -0.382  0.181 -0.345        -0.338  0.111  0.377  0.534 -0.169
## shot.put   -0.152 -0.565 -0.283         0.221  0.319 -0.489  0.407
## high.jump          0.381 -0.609                0.459        -0.435  0.134
## x400m       0.506        -0.147                0.117                -0.827
## x110mh      0.489        -0.120        -0.269         0.252  0.403  0.340
## discus     -0.108 -0.552 -0.235  0.311  0.256 -0.116  0.602 -0.282
## pole.vault -0.239  0.413  0.150  0.129  0.714         0.143  0.306
## javelin                   0.254  0.885 -0.217  0.245 -0.190
## x1500m      0.228  0.149 -0.496  0.295  0.161 -0.610 -0.321         0.136
##            Comp.10
## x100m       0.547
## long.jump   0.332
## shot.put   -0.143
## high.jump  -0.237
## x400m
## x110mh     -0.575
## discus
## pole.vault -0.323
## javelin
## x1500m      0.262
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## SS loadings       1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
## Cumulative Var    0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9
##                Comp.10
## SS loadings        1.0
## Proportion Var     0.1
## Cumulative Var     1.0
```

For each of components 1 through 3, see which events have a large (in absolute value) loading on that component. You will have to draw the line somewhere. What you want to get is a sense of what each component "mainly" contains, some things that it makes sense would go together. With that in mind:

1. 100m, 400m and 110m hurdles. Maybe also long jump, with the opposite sign. This is running fast (sprinting). The reason long jump has the opposite sign is that a lower *time* is good, but a higher *distance* jumped is better. That is to say, a decathlete that is good at one of the short running races is likely to be good at the others, and also good at long jump. (There is a long history of good sprinters also being good long jumpers, one example being Carl Lewis, who was world champion at 100m, 200m and long jump.)

2. Shot put and discus: that is, throwing heavy things. If you want, also high jump and pole vault, with the opposite sign, which says that decathletes who are good at shot put and discus tend to be bad at high jump and pole vault, and vice versa. From that point of view, this component is a contrast between throwing heavy things and getting over a

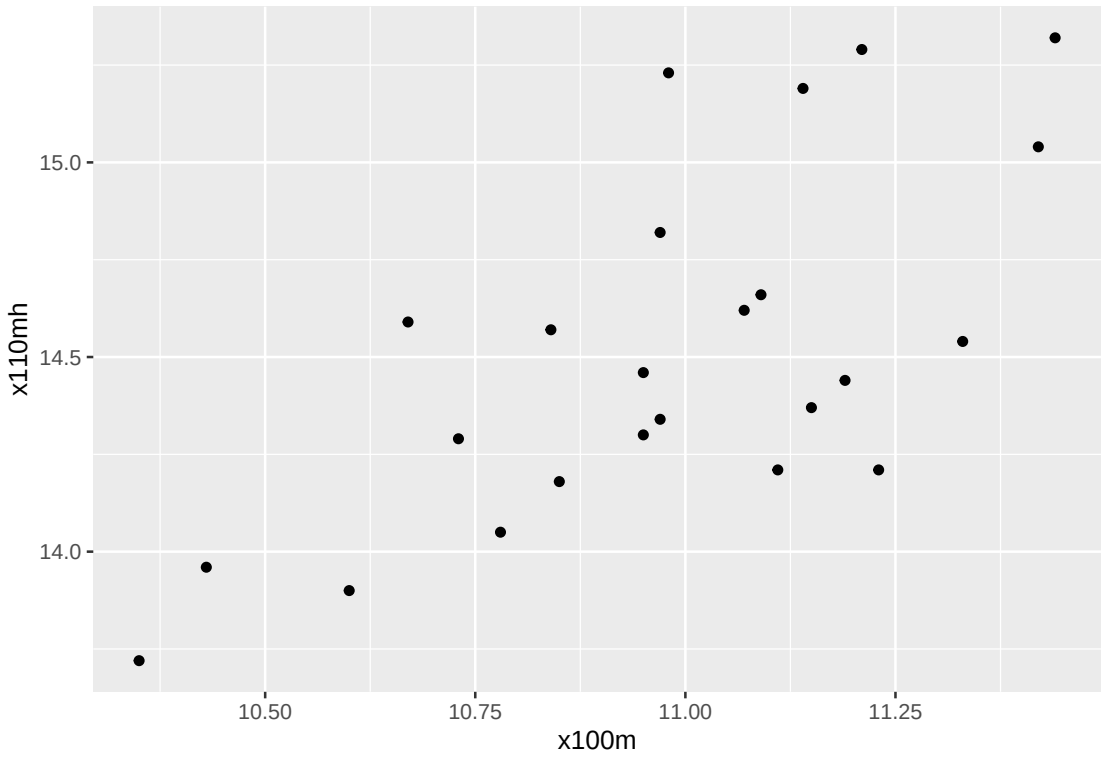
high obstacle, saying that decathletes tend to be somewhere along this dimension.

3. High jump and long jump, and 1500m the opposite way (because a high distance but a low time are good). You might describe this as something like explosiveness vs. endurance, since jumping requires a quick burst of power to jump long or high, and distance running, well, doesn't: the whole point of distance running is that you keep putting one foot in front of another, again and again.

Extra: to explore some of these things, we can make some scatterplots.

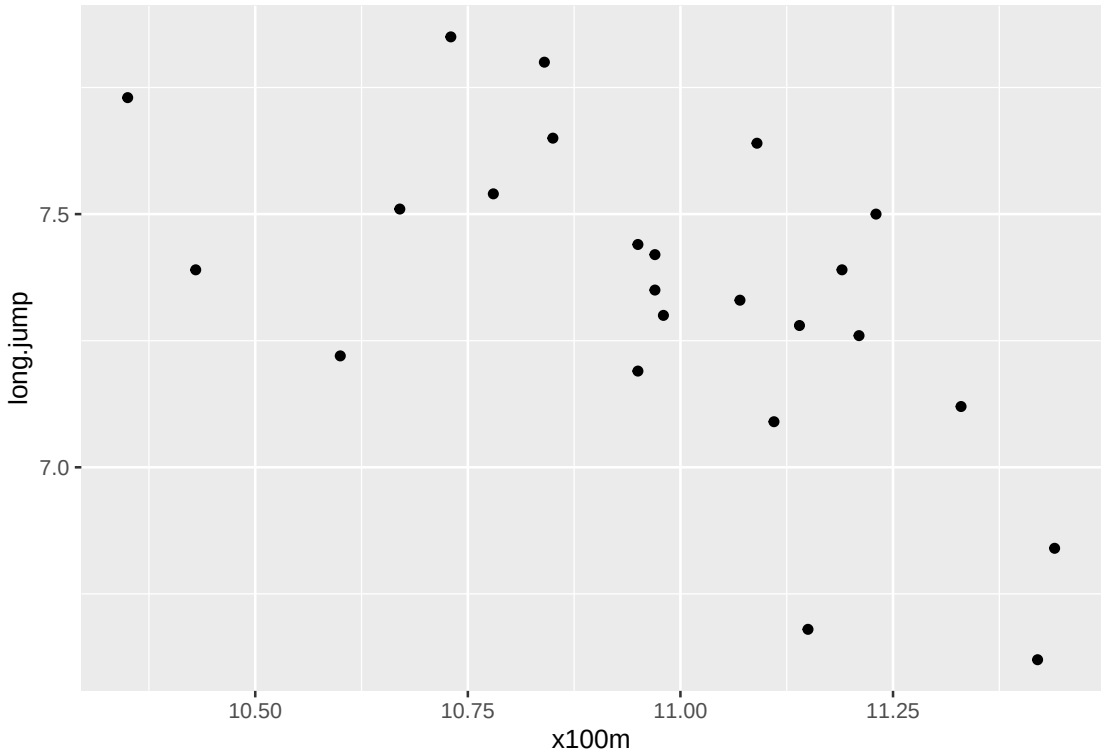
For 1, 100m vs 110m hurdles:

```
ggplot(decathlon, aes(x = x100m, y = x110mh)) + geom_point()
```



a positive correlation, and

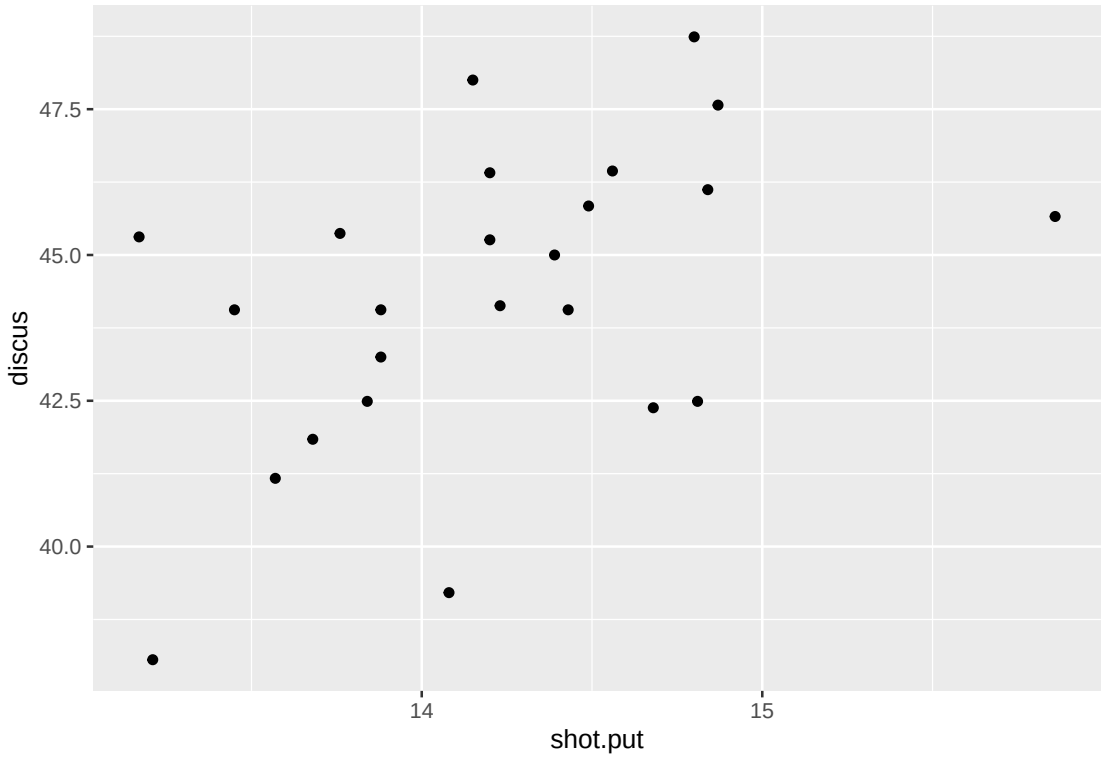
```
ggplot(decathlon, aes(x = x100m, y = long.jump)) + geom_point()
```



a positive correlation that looks like a negative one, because decathletes who jump far (a long distance) also tend to run fast (a *low* time).

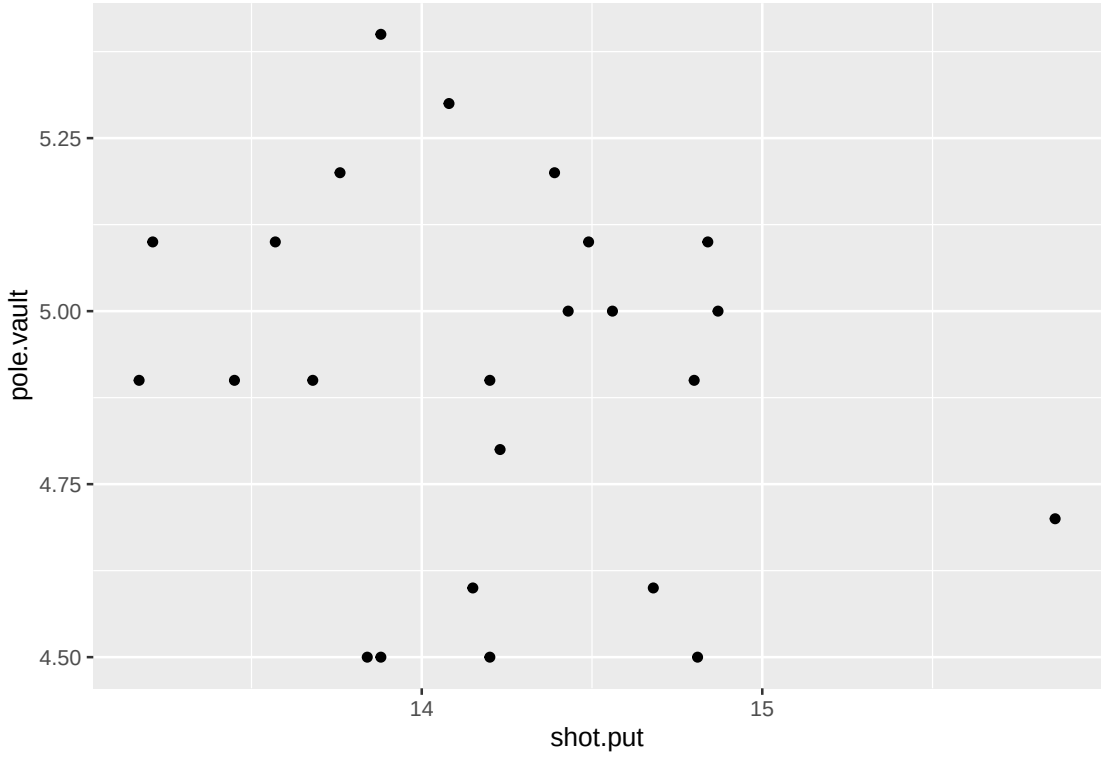
For 2., shot put and discus:

```
ggplot(decathlon, aes(x = shot.put, y = discus)) + geom_point()
```



a positive correlation; also, shot put and pole vault:

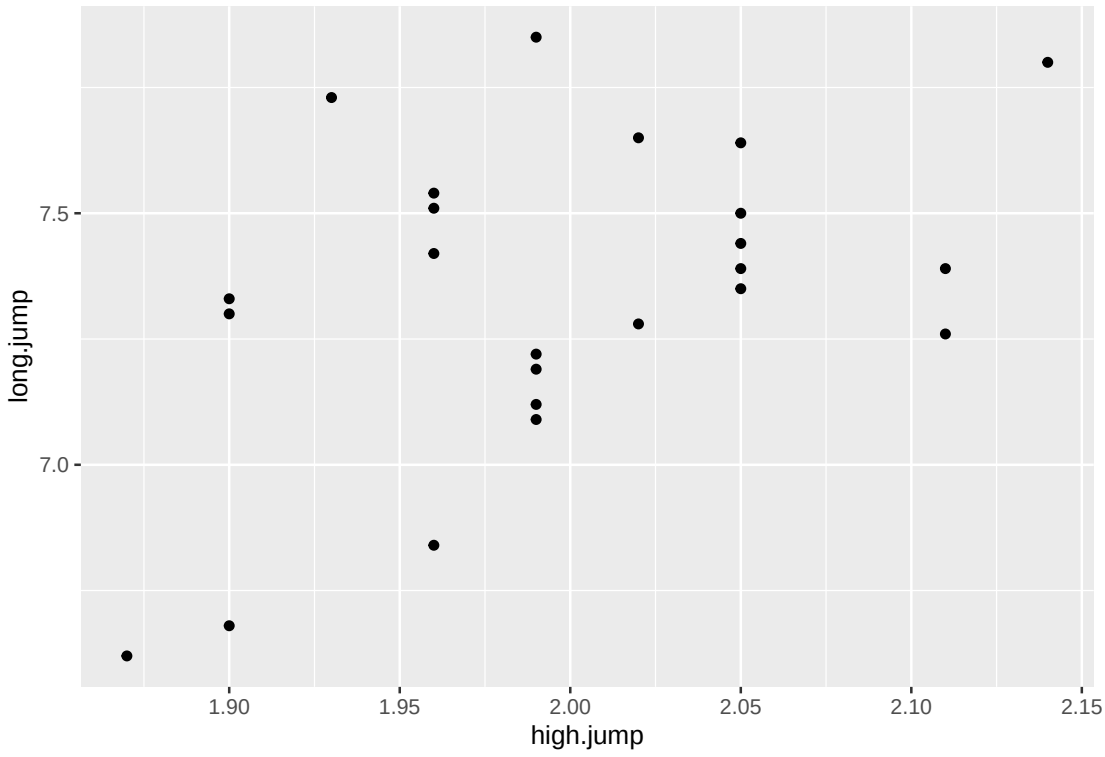
```
ggplot(decathlon, aes(x = shot.put, y = pole.vault)) + geom_point()
```



a negative correlation, though not a terribly strong one.

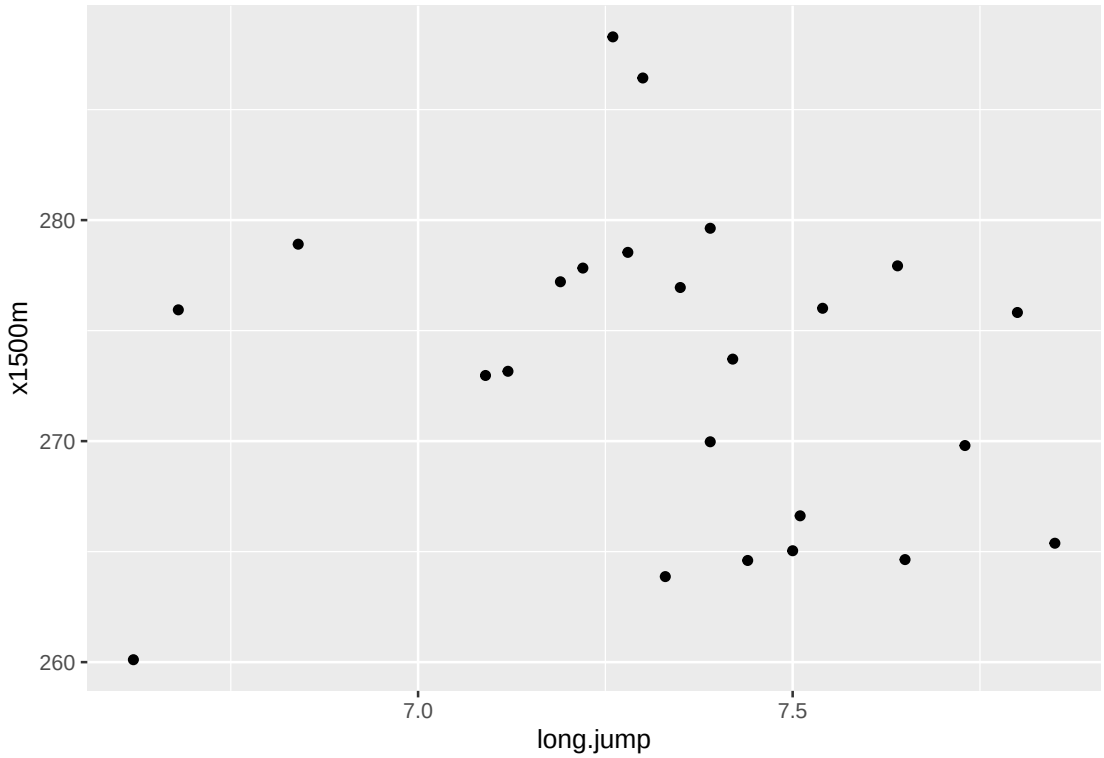
For 3., high jump and long jump:

```
ggplot(decathlon, aes(x = high.jump, y = long.jump)) + geom_point()
```



a positive correlation, and finally long jump and 1500m

```
ggplot(decathlon, aes(x = long.jump, y = x1500m)) + geom_point()
```



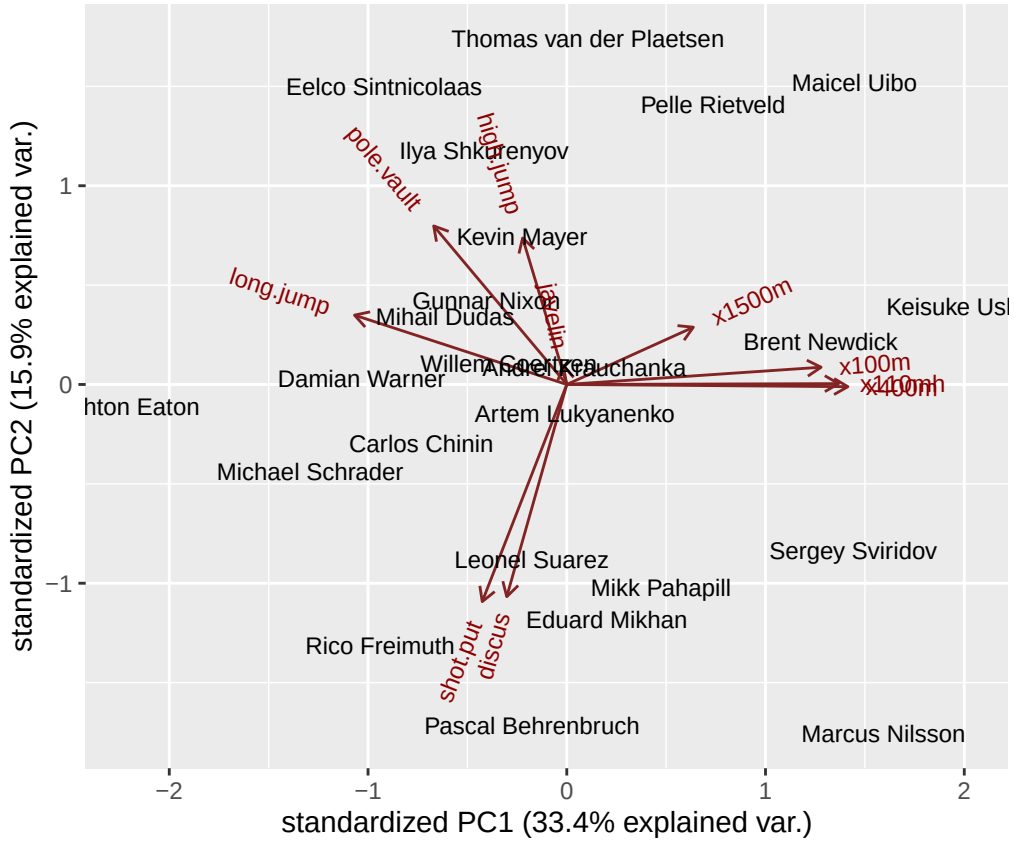
This is actually not a very convincing correlation at all. This is supposed to be a negative correlation that looks like a positive one, since someone who jumps a long way is, we thought, likely to take a long time to run the 1500m (slow).

(f) Make a biplot of the principal component analysis, including the names of the decathletes on your biplot.

Solution:

This is much most easily done with `ggbiplot` from the package of the same name:

```
ggbiplot(decathlon.1, labels = decathlon$name)
```



The only difficult thing here is labelling the decathletes; use `labels` with a column that contains their names, such as the one in the original dataframe.

I think you can construct a biplot from the output plus `ggplot`, but this is much harder work than necessary.

(g) Which events do you think Pascal Behrenbruch is particularly strong or weak at? What about Ashton Eaton? By looking at the original data, see whether you were right. Hint: if you look at the original dataframe using `View`, you can sort the rows by any variable by clicking on the little arrows next to the variable's name. You won't easily be able to grab this to hand it in, so describe what you found.

Solution:

Pascal Behrenbruch is at the bottom of the biplot, near the head of the arrows for shot put and discus, so he will be good at these events. The arrow for high jump points the other way, so this ought to be a weaker event for him. In the data, he was the best shotputter (the only one to break 15m), and the eighth best discus thrower (out of 24); his high jump was in the middle, neither good nor bad.

Extra: Behrenbruch has a Wikipedia page, from which I learned that he is tall (6 ft 5 in) but also heavy (over 200 pounds, which seems heavy for a decathlete). It therefore makes sense that he would be good at the strength events.

What about Ashton Eaton? He is over on the left, so he will be good at long jump (at the

head of the arrow). On the face of it, it looks as if he should be bad at the running events, but being at the wrong end of the arrow means that his times are *low* and therefore that he is *good* at them as well. (Note that his component 2 score is close to zero, so that he is average at the things that point up and down, neither good nor bad.) In the data: his 100m is the best, 110m hurdles is also best, 400m is best as well, and long jump is 3rd best. That seems pretty convincing.

Extra: it is perhaps not surprising that Eaton was the world champion in 2013; being best in several events and bad at none of them is a good recipe for success at decathlon. The silver medallist was Michael Schraeder, another good runner, not quite as good as Eaton, but a bit better at shot put and discus. The bronze medallist was the Canadian Damian Warner, mid-left of the biplot, so with a similar profile to Eaton. The best decathlete with a different profile was Eelco Sintnicolaas: he was better at pole vault (2nd) and high jump (12th), and worse at shot put (10th from bottom) and discus (2nd worst), the opposite to Behrenbruch. The left-right position of Sintnicolaas compared to Behrenbruch suggests that the former was better at running, which would explain his better overall standing.

All the results are here. For the dataset, I omitted the athletes that did not take part in all ten events, to avoid missing values.