DATA 608 Meetup 10: Dimensionality Reduction

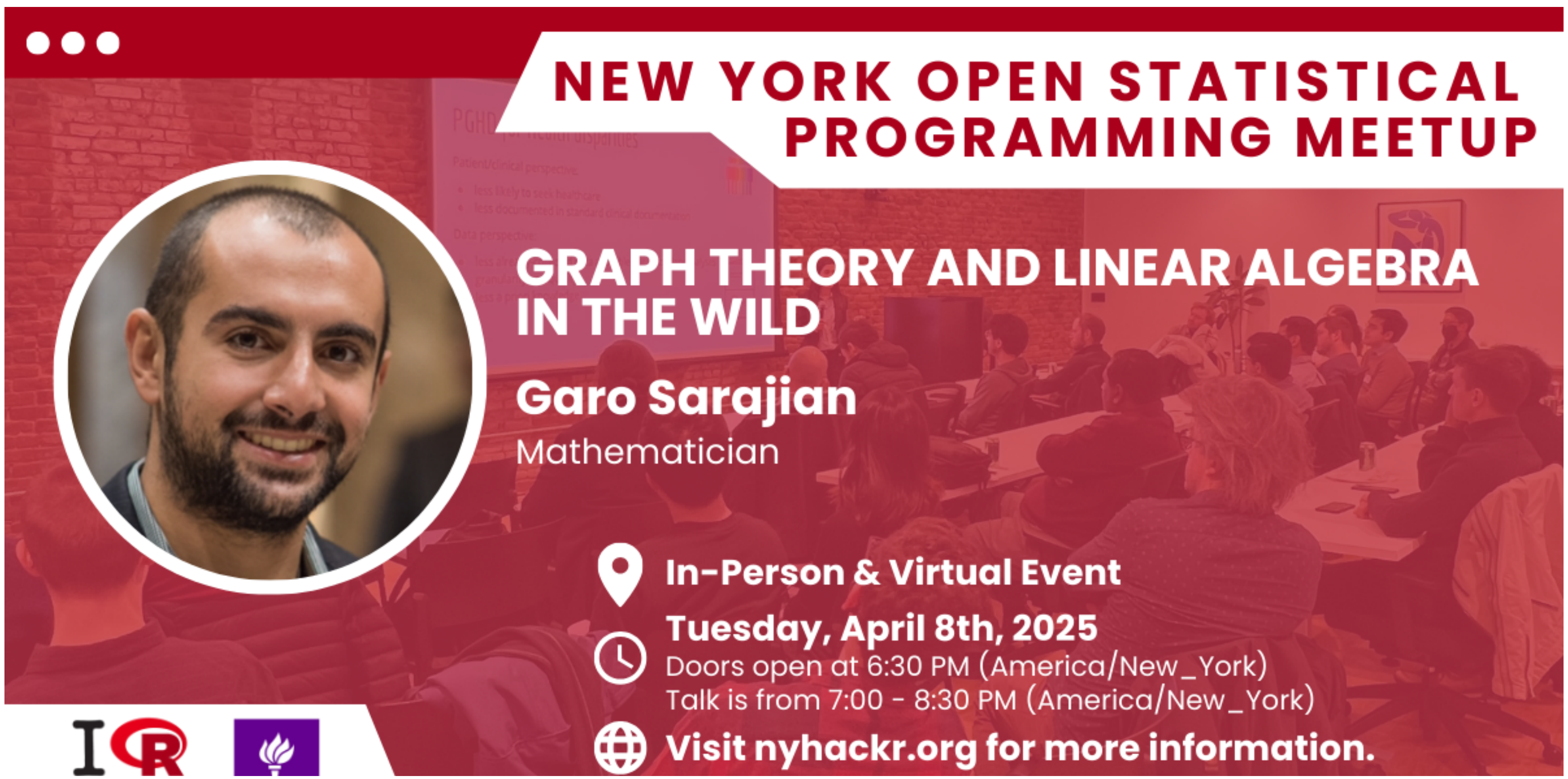
George I. Hagstrom

Week Summary

- Data Story 5 is Due this Sunday at Midnight
- Reading is Section 12.3 of Fundamentals of Data Visualization
- What to do when the data has too many variables to visualize?

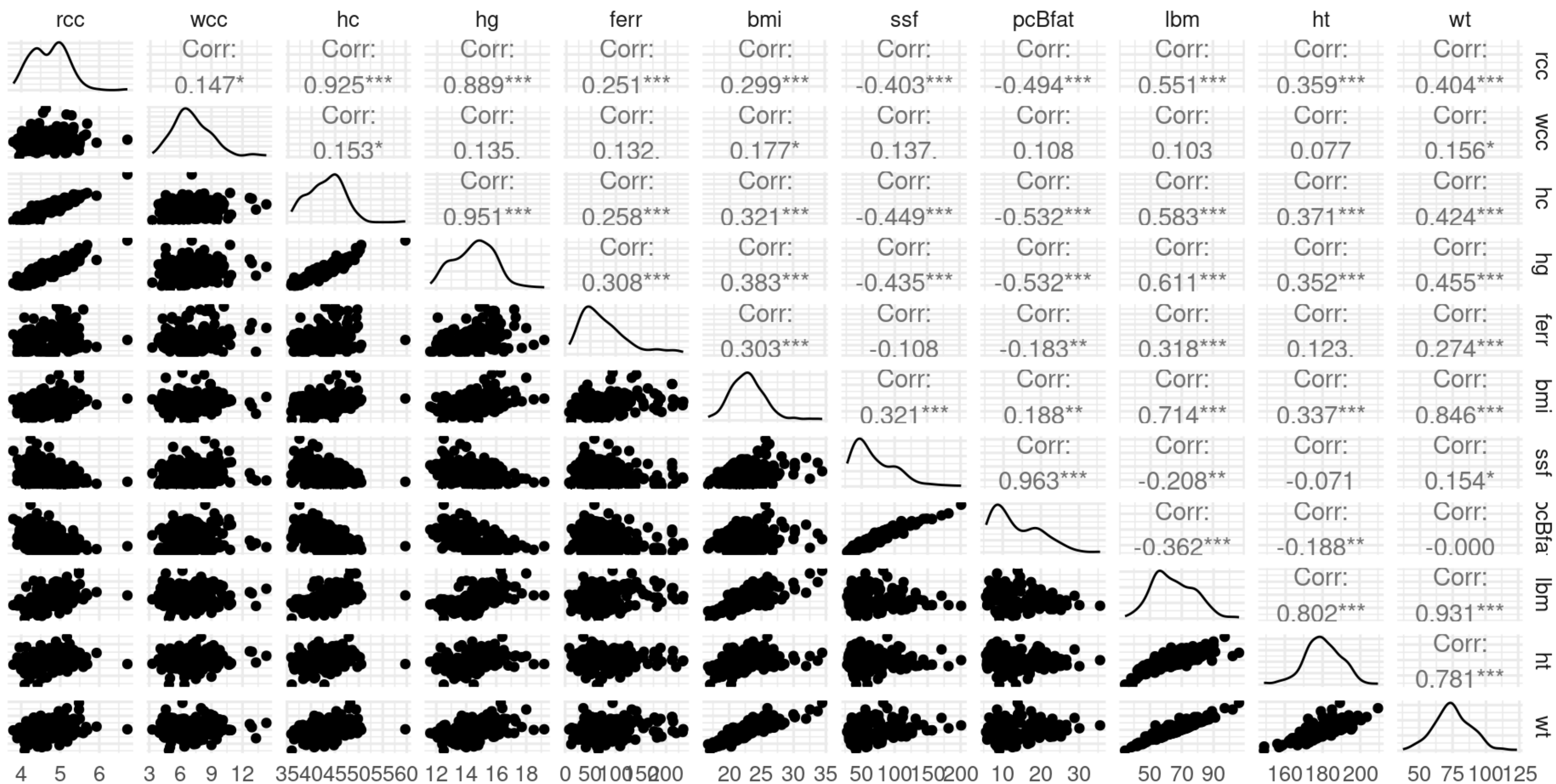
NY Open Statistical Computing Meetup

- April 8th 6:30-8:30+
- Great Networking/Learning Opportunity



Many variables?

- Pair Plot?



Pair Plots are Limited

- Data is physiological properties of Australian Olympians
- Takes time to read
- Other options like correlograms also somewhat limited
- We could use an alternative even though these techniques are useful

Hopeless when data increases more

- Here is a dataset of biomarker measurements from both ovarian cancer patients and healthy controls
- There are 4000 variables
- There will be no pair-plots

```
# A tibble: 215 × 4,000
   `0.063915` `0.033242` `0.018484` `0.0086177` `0.035629` `0.037925`
`0.028865`
        <dbl>      <dbl>      <dbl>       <dbl>      <dbl>      <dbl>
<dbl>
 1     0.0254     0.0511    0.0563      0.0217      0.0274     0.0149
0.0225
 2     0.0255     0.0361    0.0542      0.00974     0.0275     0.0523
0.0428
 3     0.0128     0.0297    0.0793      0.0507      0.0397     0.0577
0.0445
 4     0.0198    -0.0106   -0.00750     0.0190      0.0688     0.0618
0.0390
```

5	0.0390	0.0394	0.00134	0.0262	0.0441	0.0440

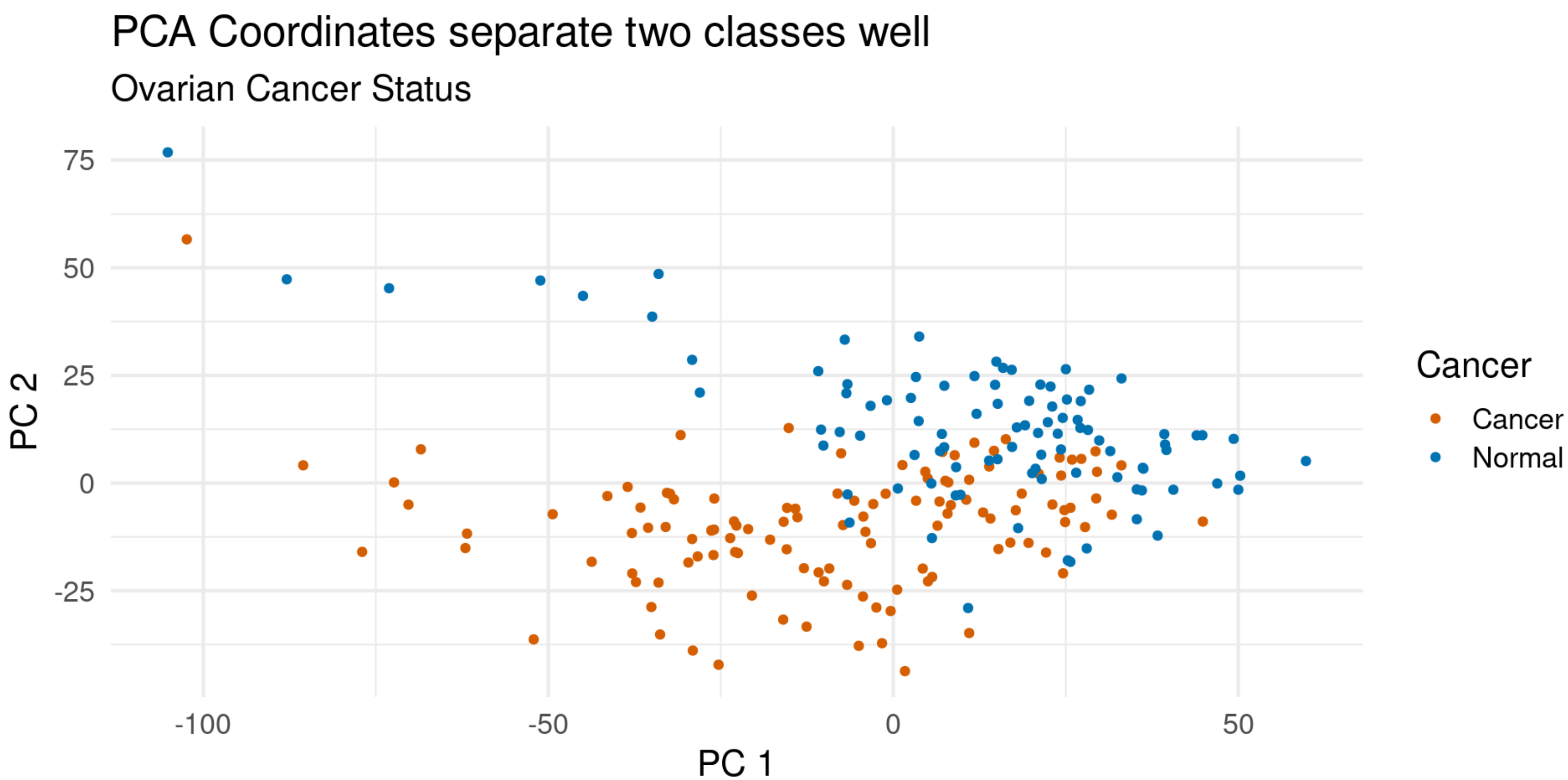
0.0396

Dimensionality Reduction

- Many datasets have a large number of variables
- We often have intuition that the data follows simpler patterns
- Otherwise we couldn’t understand it
- This motivates a search for simpler representations, which can be powerful visualization tools

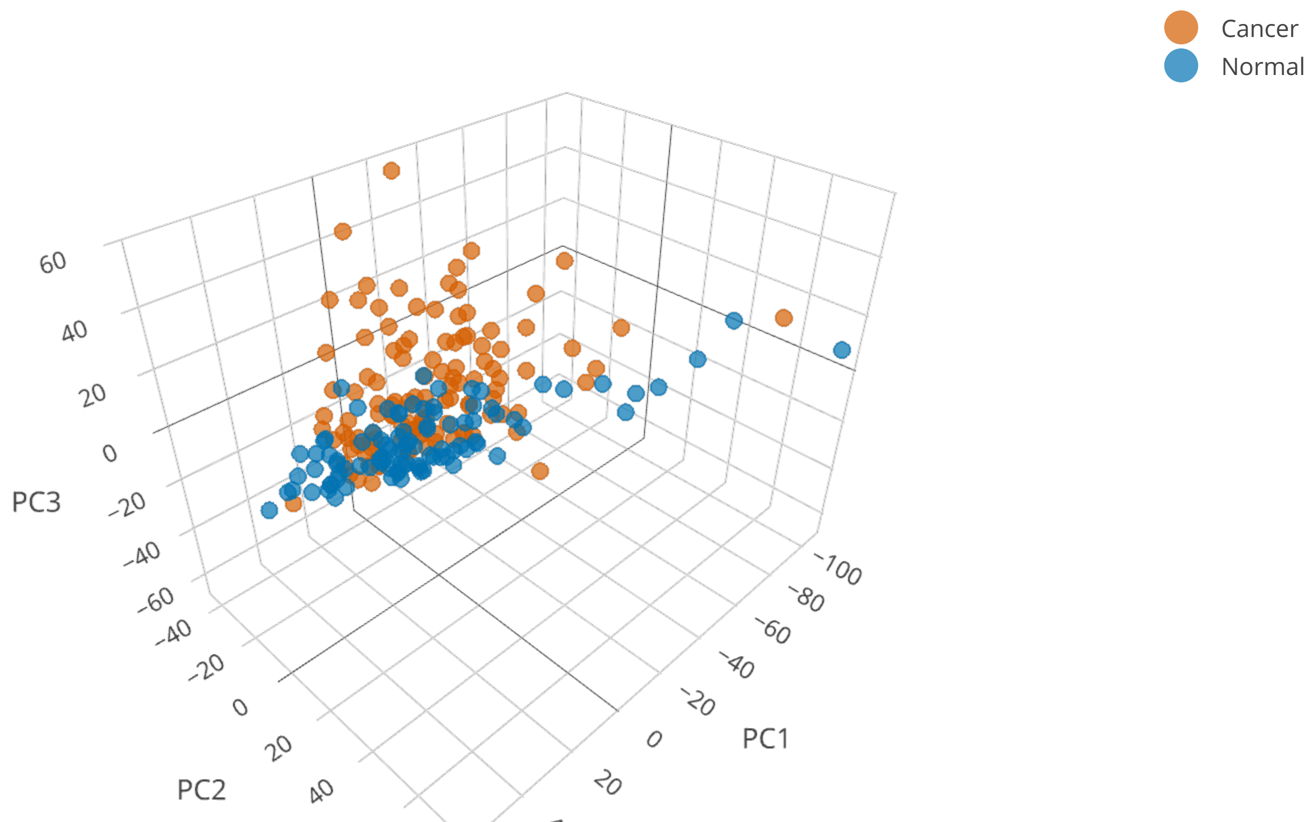
Low-D Representation

- Using a tool called Principal Component Analysis, can separate the groups neatly



3D Representation

- 3D Neatly separates them

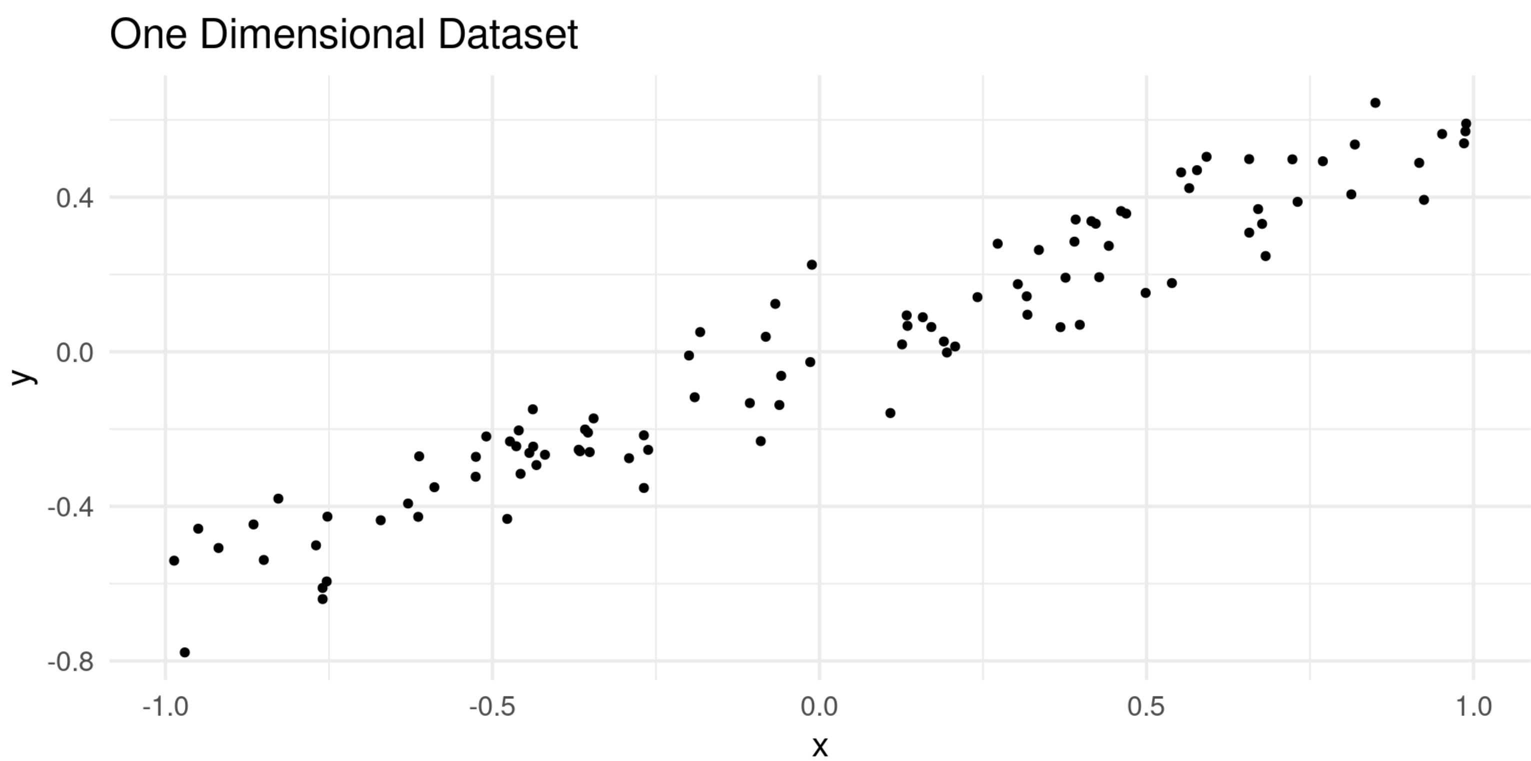


Principal Component Analysis (PCA)

- PCA finds a new coordinate system
- New directions called `Principal Components`
- Each PC is a combination/rotation of other ones
- PCs ordered based variability of the data

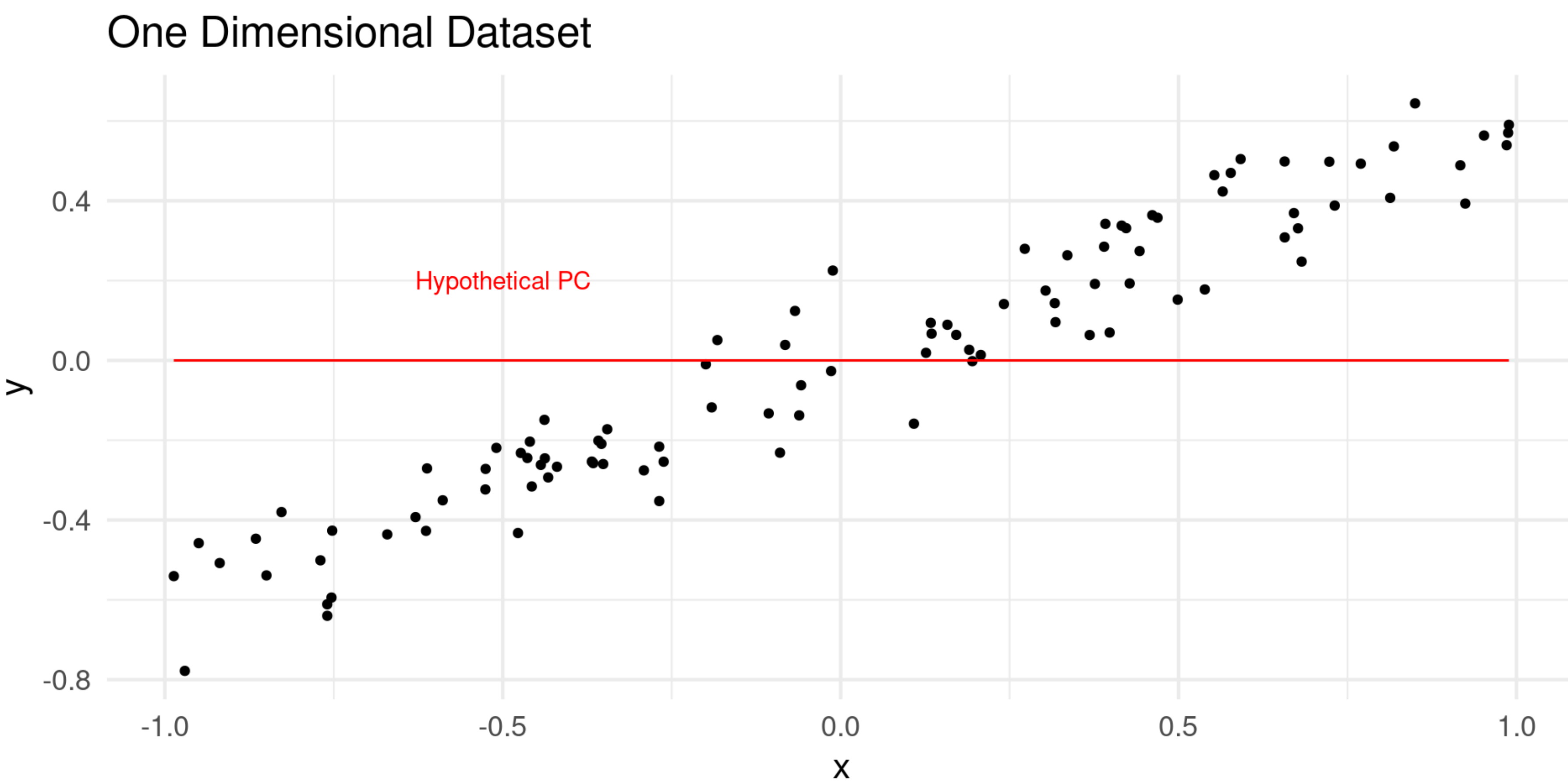
Simple PCA example

- Make a “1D” dataset in two dimensions
- Data lies along a straight line with some noise



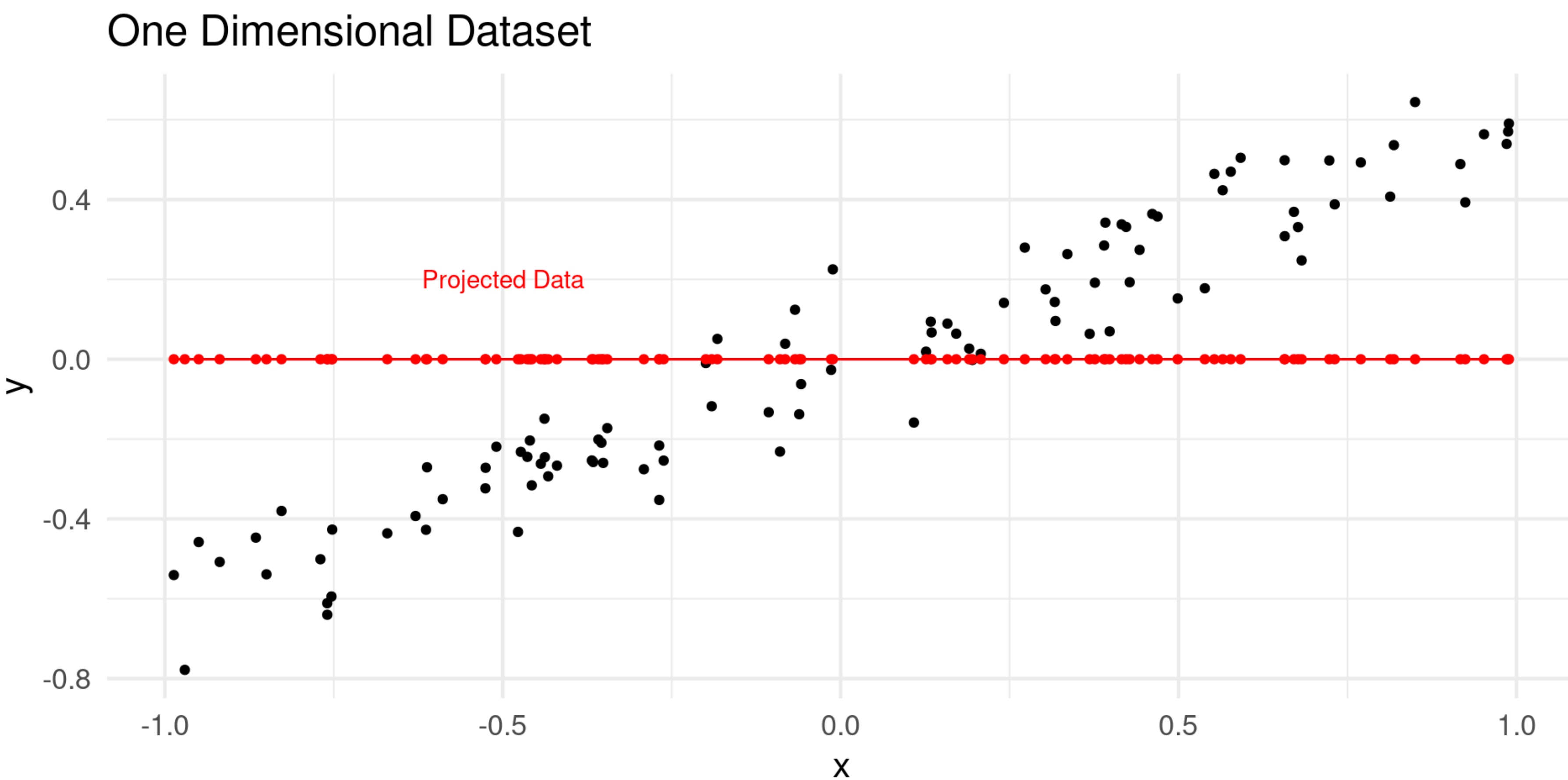
Simple PCA Example

- PCA looks for vectors along which the “variance” is maximized



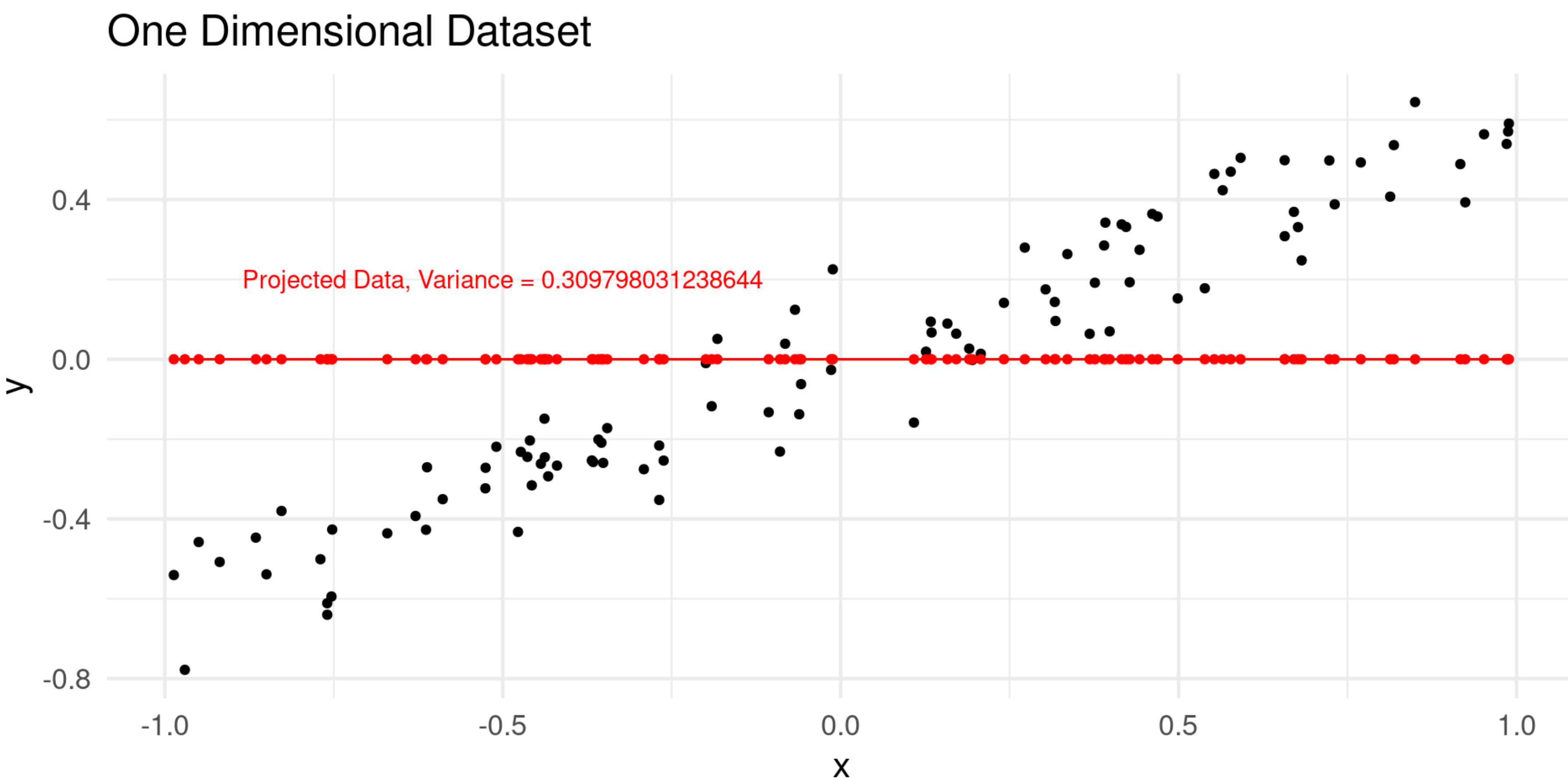
Simple PCA Example

- Project data points onto component



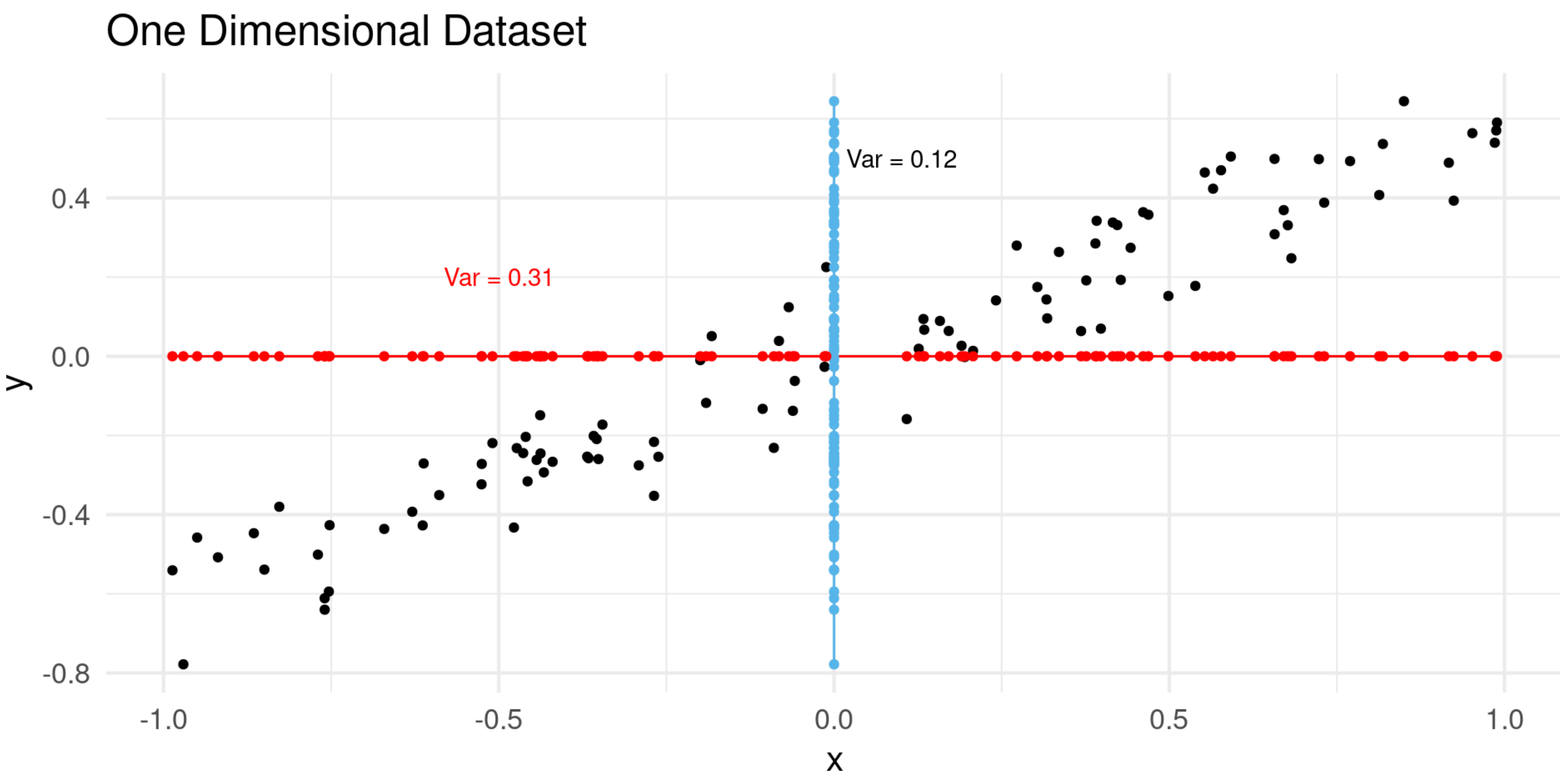
Find the Variance

- Project data points onto component



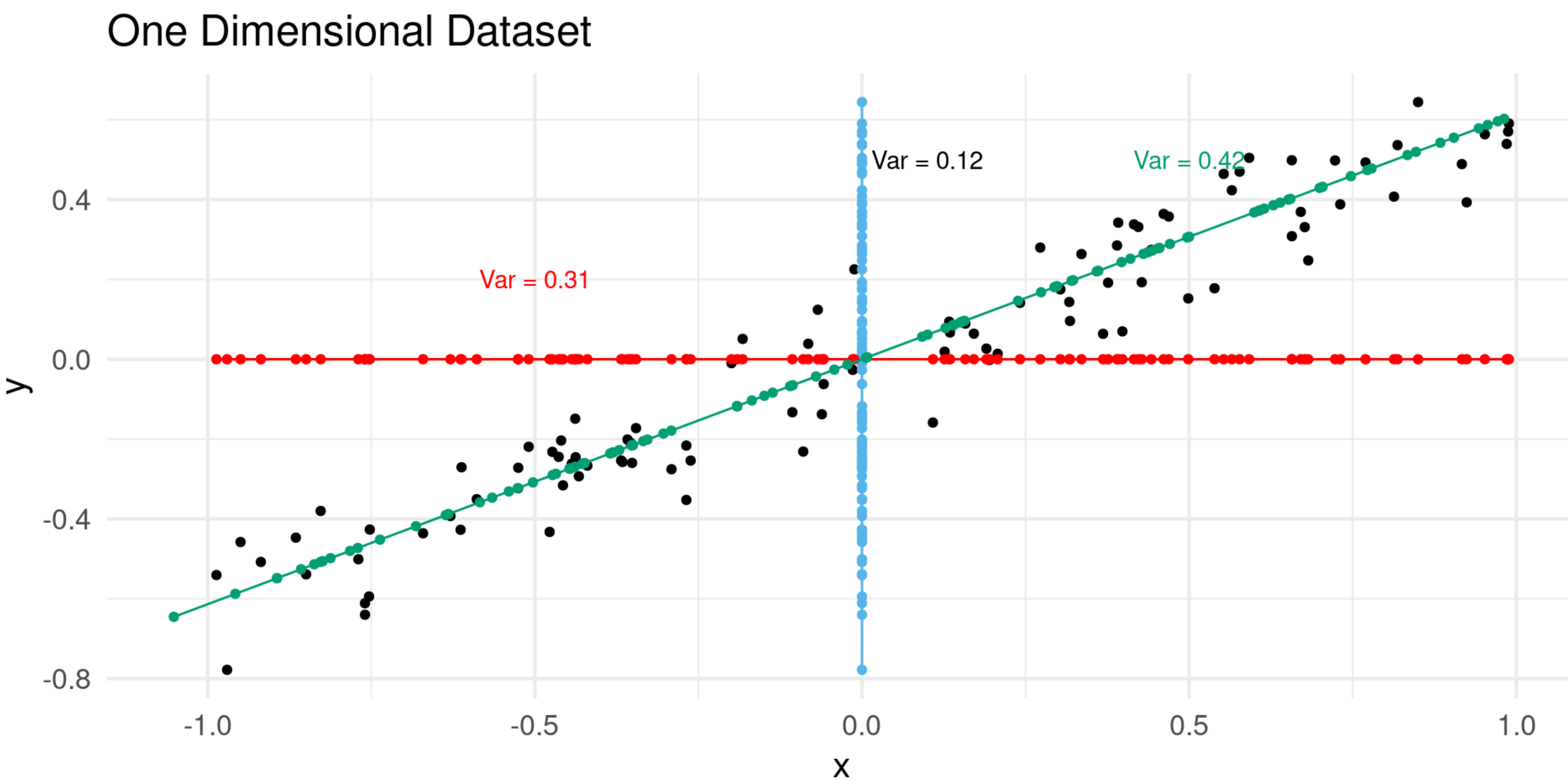
Looking at Multiple Projections

- Any projection is possible



First Principal Component

- PC1 maximizes variance

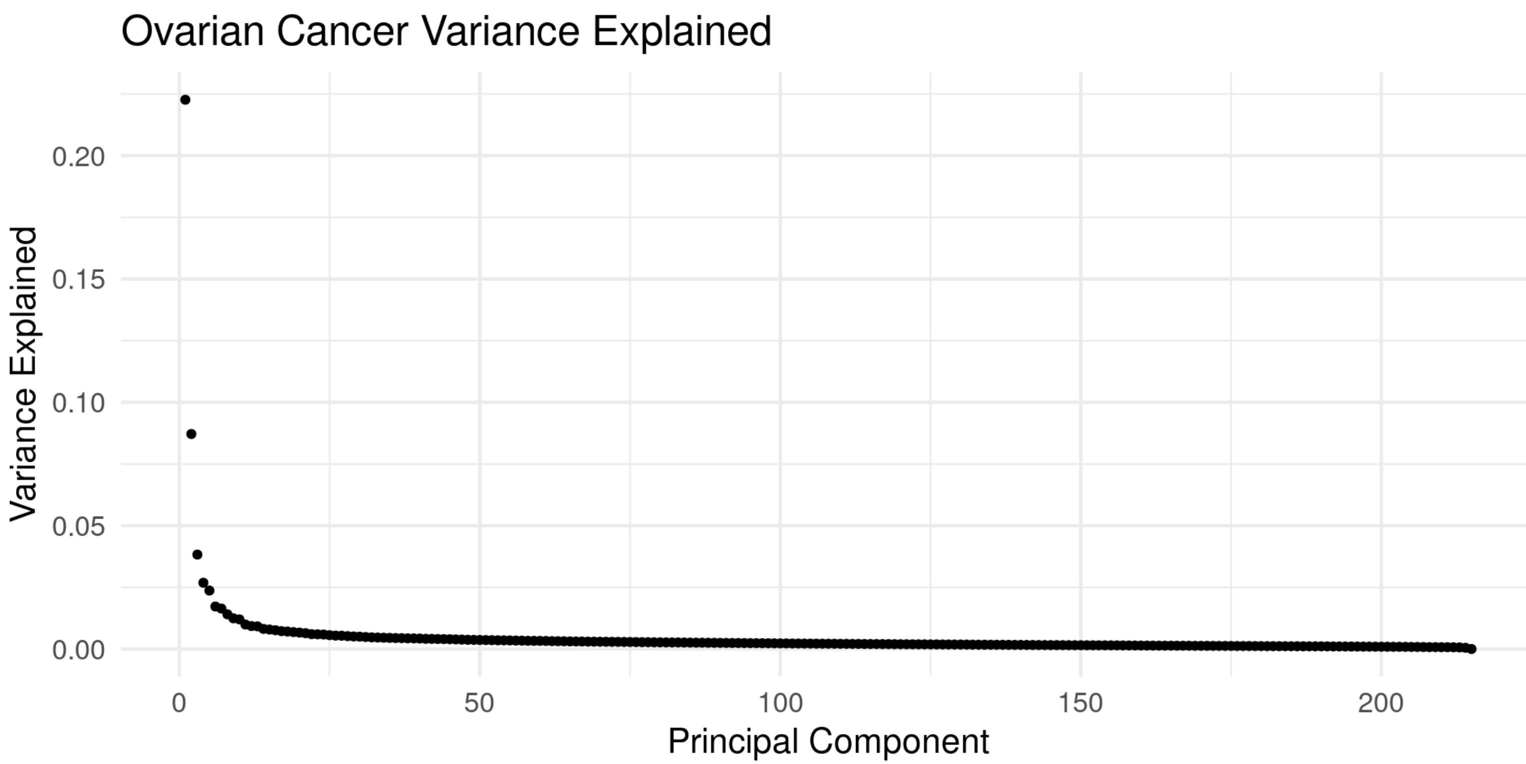


Subsequent Projections Repeat Process

- PC1 is the direction with the most variance
- PC2 is perpendicular to PC1, most variance
- PC3 is perpendicular to PC1 and PC2, most variance
- N dimensional dataset has N PCs

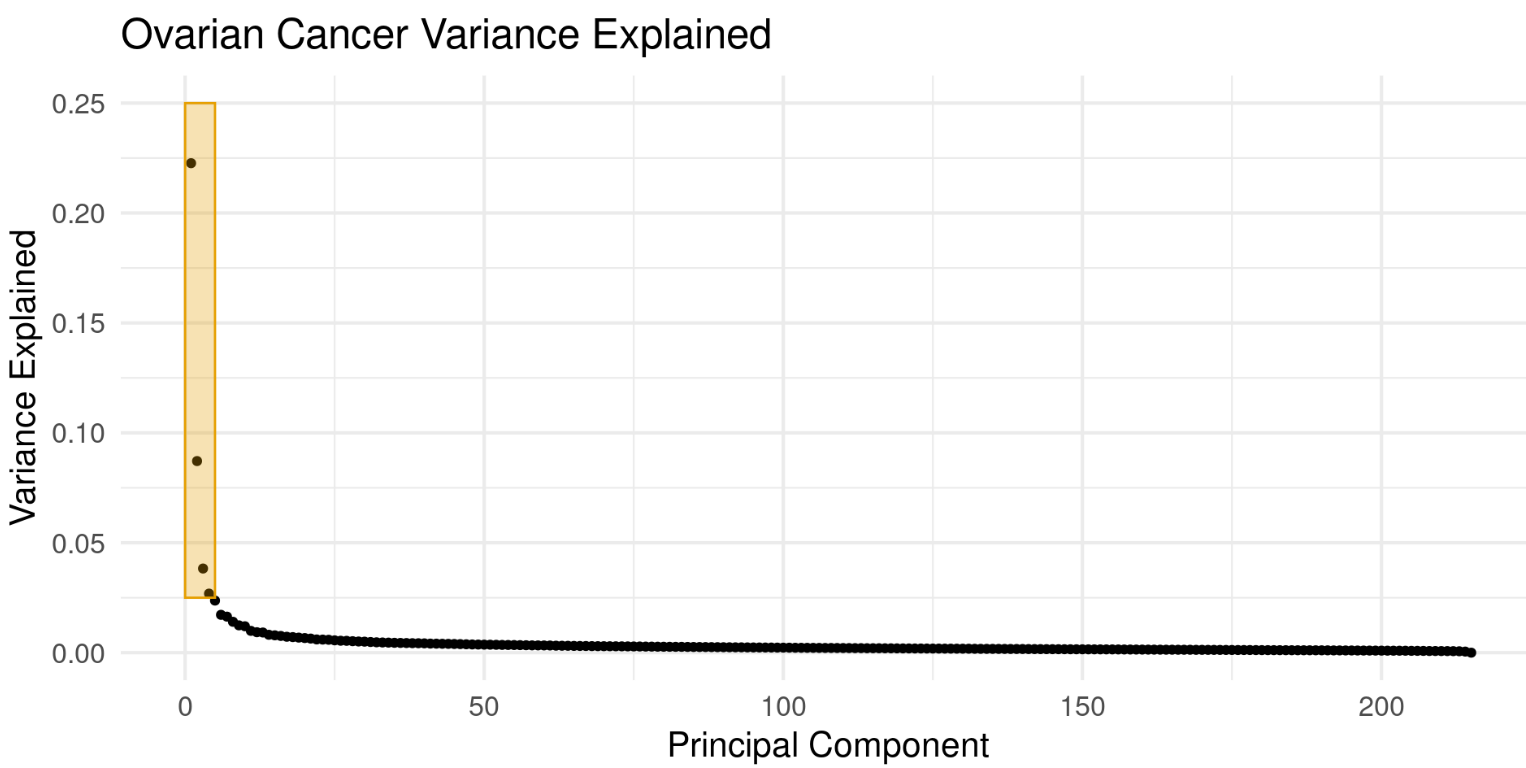
Why is it Dim. Reduction?

- Each PC is a variable
- You can discard low variance PCs

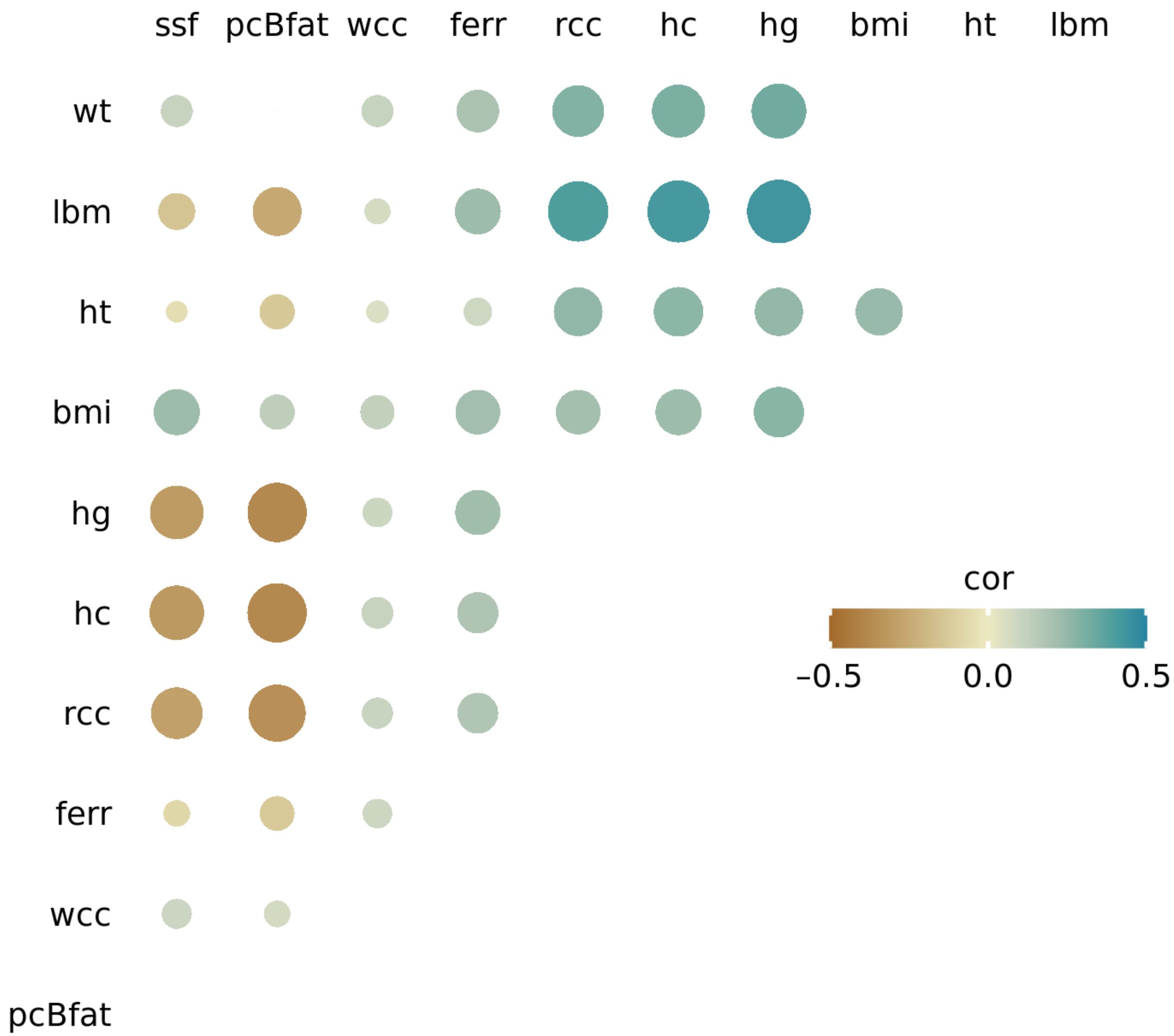


Picking a Dimension

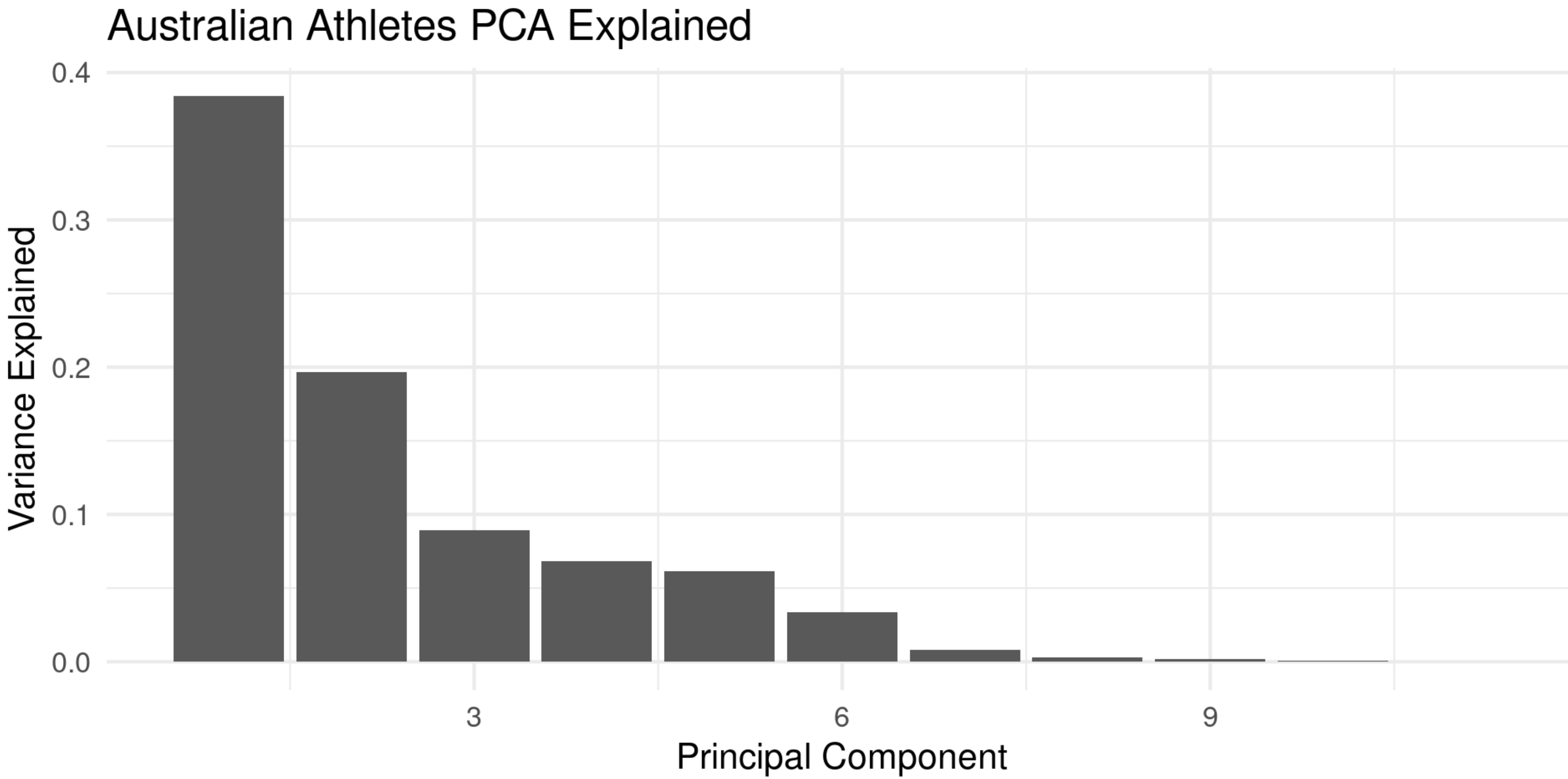
- You can discard low variance PCs
- Use “Elbow” of plot to pick cutoff



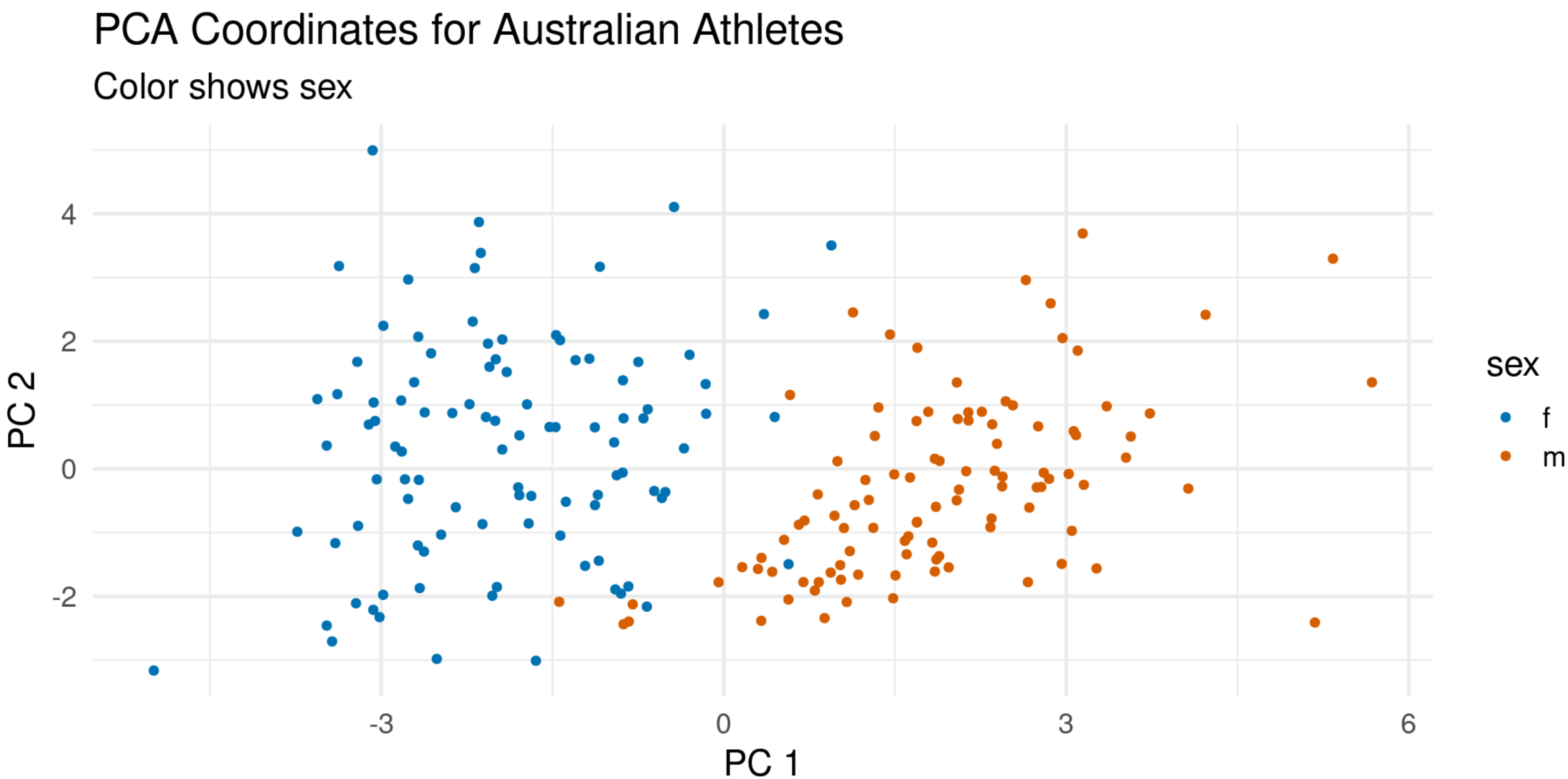
Australian Atheletes Example



Athletes PCs

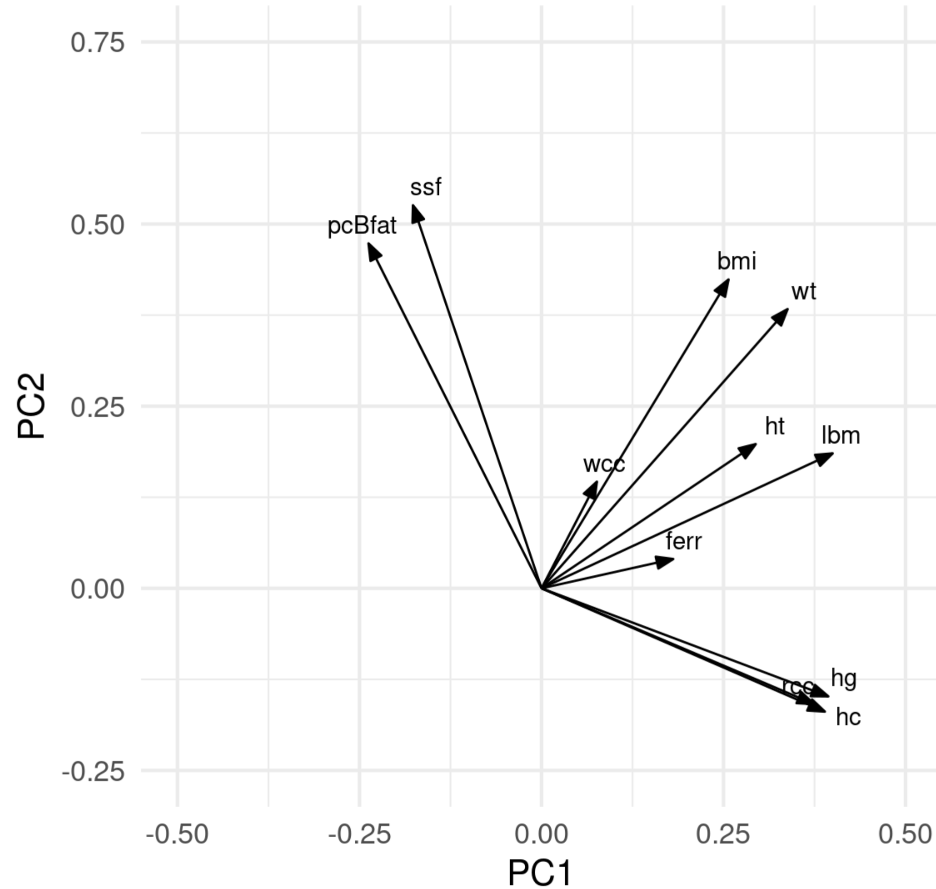


PCA Clusters

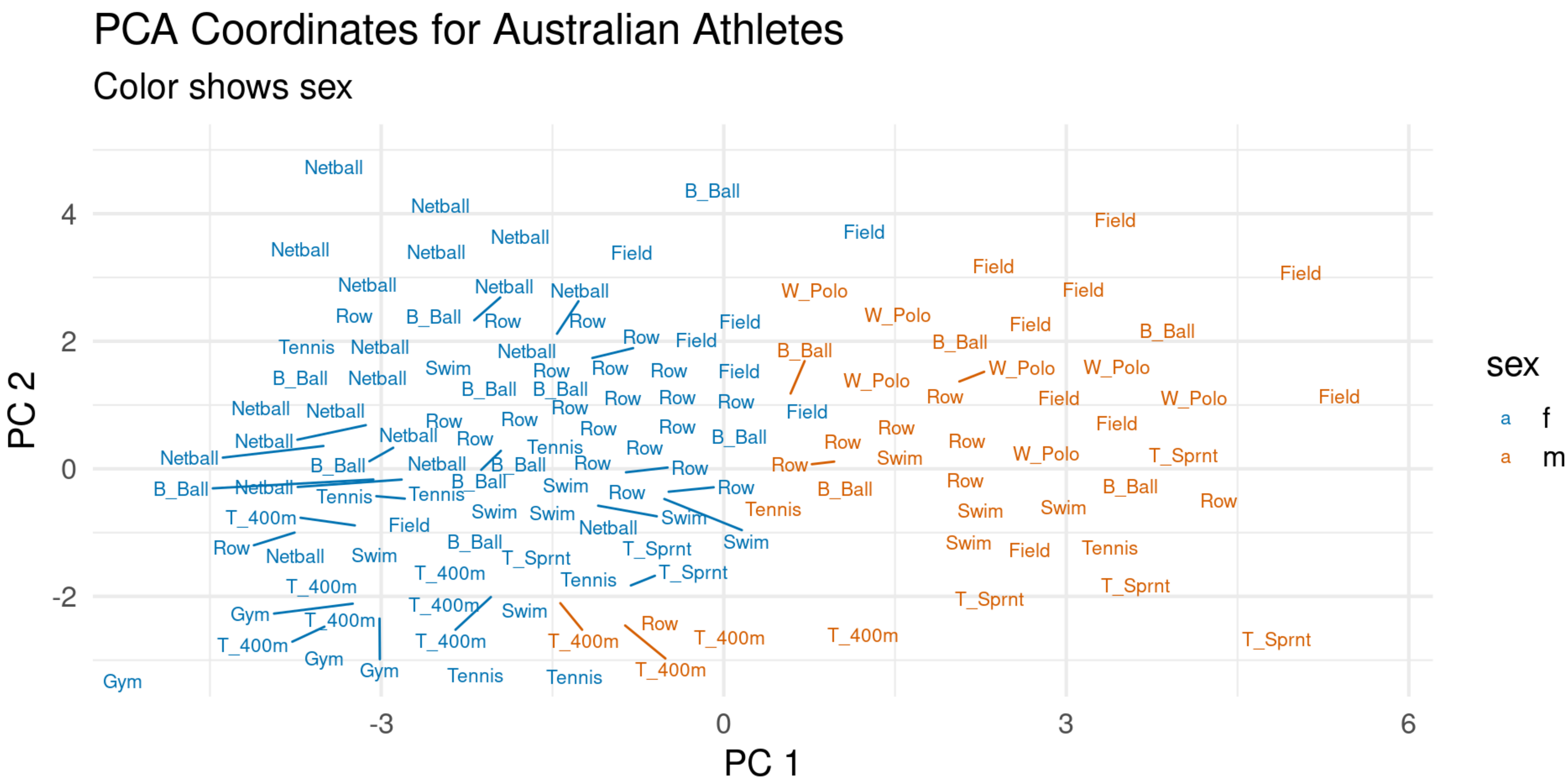


PC Clusters

- PC1: +Size, -Leanness (Sex)
- PC2: +Size, -Red Blood Cells



Sports

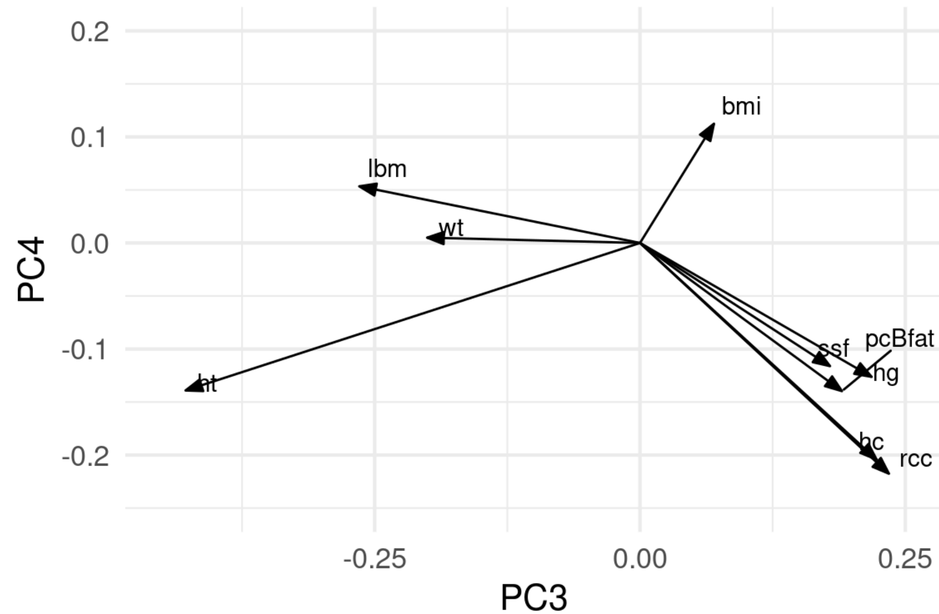

PCA Coordinates for Australian Athletes
Color shows sex
PC 1
PC 2
sex
f
m

Sports Clusters

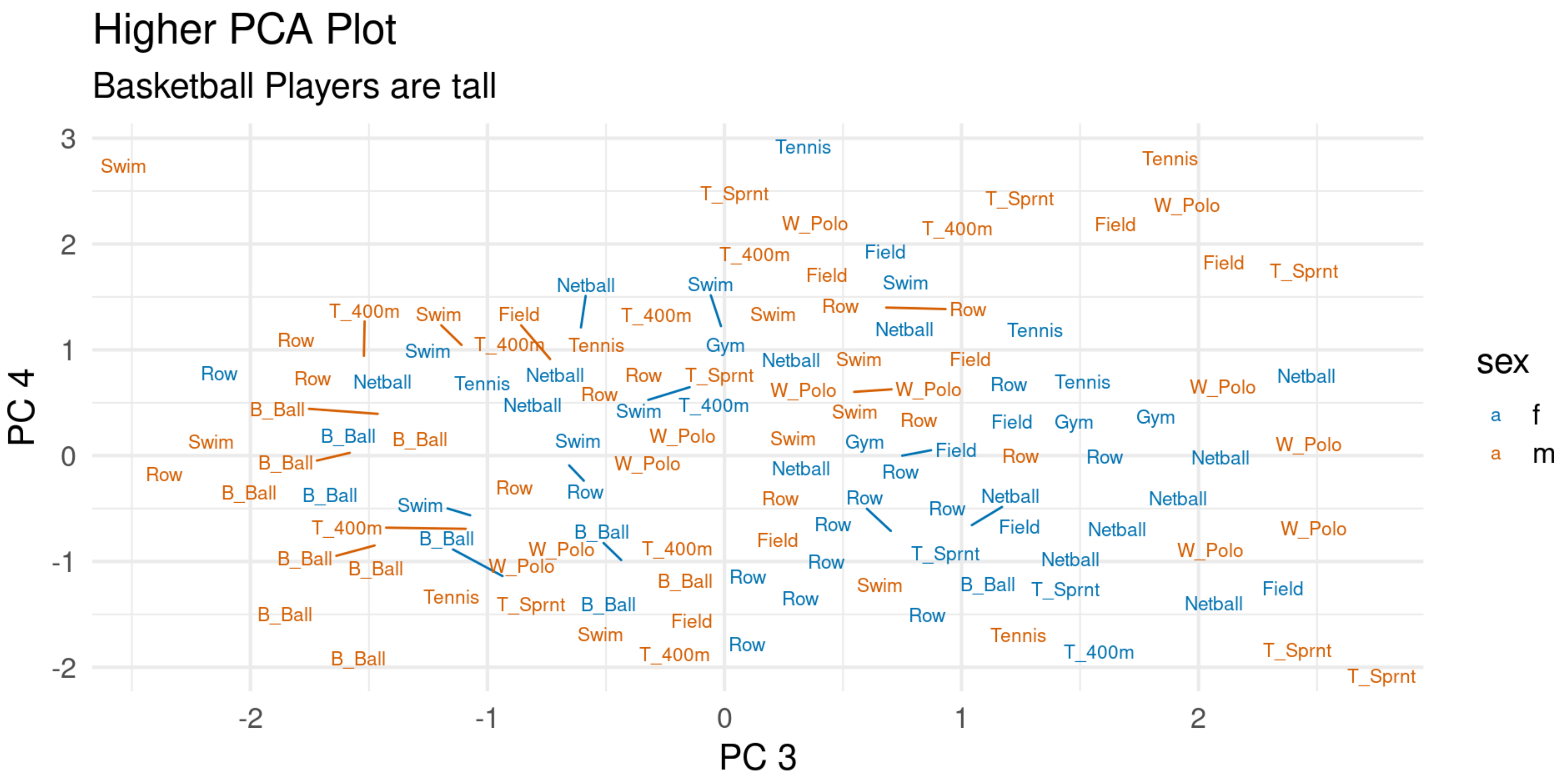
- 400m, gymnasts, tennis players have low PC2
- Netball, Field have high PC2
- Rowers, Sprinters, Swimmers, Water Polo in the middle

Higher PCs Contrast

- PC3: +Body Fat, RBCs, -Height
- PC4: +Height, Fat, RBCs, -BMI



Sport Clusters 2



How to make these plots in R

- `prcomp` and `scale`
- `retx` keyword calcualtes embeddings

```
aus_fit = aus_athletes |>
  select(where(is.numeric))  |>
  prcomp(scale = TRUE)

# or
# gets rid of mean and rescales variance

aus_fit = aus_athletes |>
  select(where(is.numeric))  |>
  scale() |>
  prcomp(retx = TRUE)
```

Contents of fit object

```r
aus_fit
```

```
Standard deviations (1, .., p=11):
 [1] 2.23404856 1.59923950 1.07582851 0.94294793 0.89180307 0.65872338
 [7] 0.32428594 0.20232605 0.15228831 0.07279582 0.03293771

Rotation (n x k) = (11 x 11):
               PC1         PC2         PC3          PC4         PC5
PC6
rcc     0.37449926 -0.15896859  0.23456267 -0.217592671  0.23113528
0.24042584
wcc     0.07607989  0.14643132  0.63731981 -0.193874709 -0.71219980
-0.14657450
hc      0.38921924 -0.16930334  0.22299174 -0.204335027  0.21329896
0.14848437
hg      0.39398646 -0.14839419  0.21826369 -0.126293900  0.24631047
0.04743775
```

Contents of fit object

```
str(aus_fit)
```

```
List of 5
 $ sdev    : num [1:11] 2.234 1.599 1.076 0.943 0.892 ...
 $ rotation: num [1:11, 1:11] 0.3745 0.0761 0.3892 0.394 0.181 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:11] "rcc" "wcc" "hc" "hg" ...
  .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
 $ center  : Named num [1:11] -3.77e-16 -5.22e-17 3.53e-17 -3.40e-16 -5.74e-17
...
  ..- attr(*, "names")= chr [1:11] "rcc" "wcc" "hc" "hg" ...
 $ scale   : Named num [1:11] 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "names")= chr [1:11] "rcc" "wcc" "hc" "hg" ...
 $ x       : num [1:202, 1:11] -2.07 -1.9 -3.38 -2.68 -1.8 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
```

Meaning of Components

- `sdev` is “standard deviation” of components
  - square to get “variance”,
  - divide by number of variables to convert to percentage
- `rotation` is matrix where rows are Principal Components
- `x` is matrix that contains embeddings of each data point

Tidy PCA

- Use `augment` from `broom` to make objects tidy

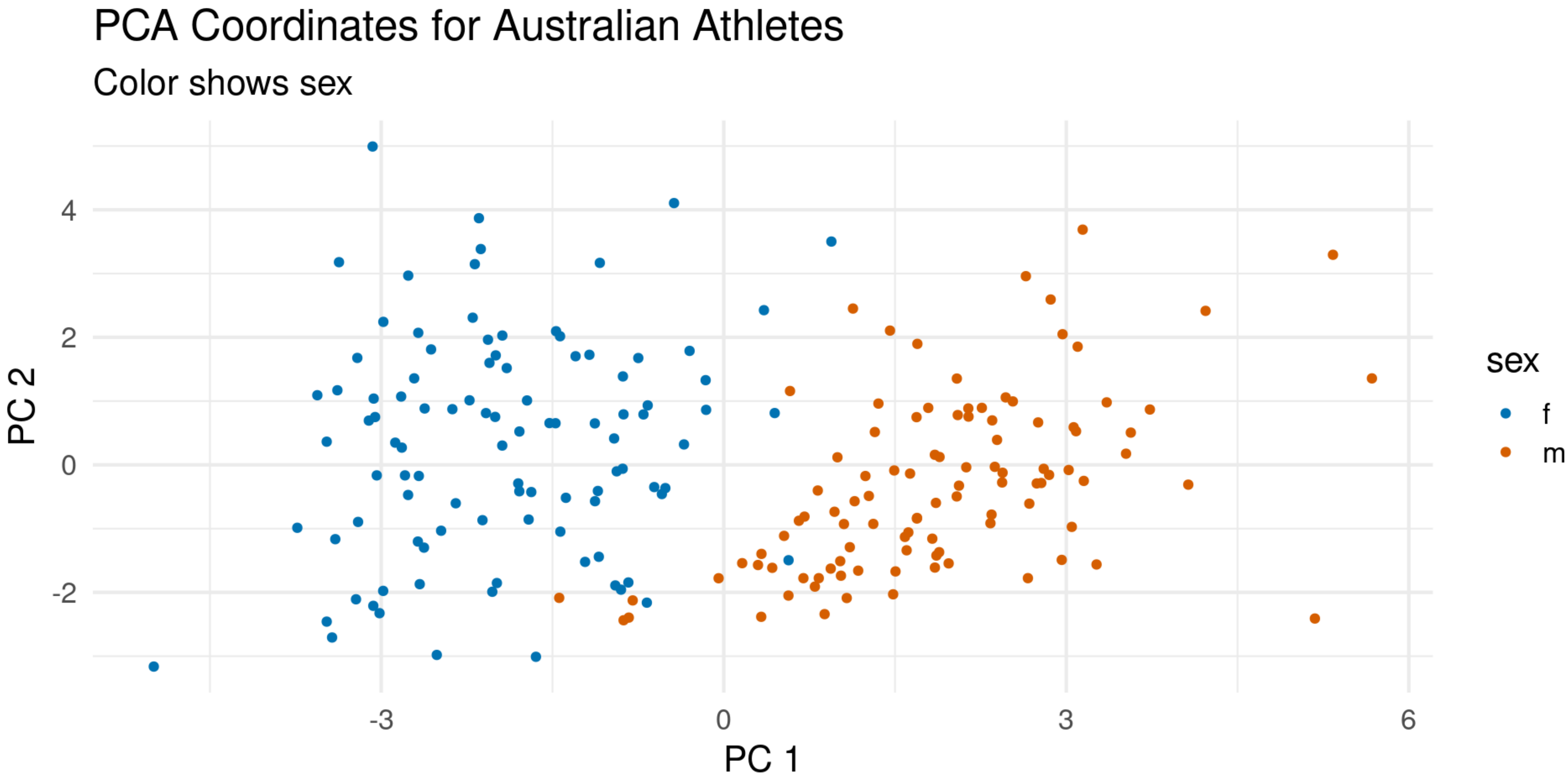
```r
aus_fit  |>
  augment(aus_athletes)
```

```
# A tibble: 202 × 25
   .rownames   rcc   wcc    hc    hg  ferr   bmi   ssf pcBfat   lbm    ht
wt
   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
<dbl>
 1 1          3.96   7.5  37.5  12.3    60  20.6 109.    19.8  63.3  196.
78.9
 2 2          4.41   8.3  38.2  12.7    68  20.7 103.    21.3  58.6  190.
74.4
 3 3          4.14   5    36.4  11.6    21  21.9 105.    19.9  55.4  178.
69.1
 4 4          4.11   5.3  37.3  12.6    69  21.9 126.    23.7  57.2  185
74.9
 5 5          4.45   6.8  41.5  14      29  19.0  80.3   17.6  53.2  185.
64.6
```

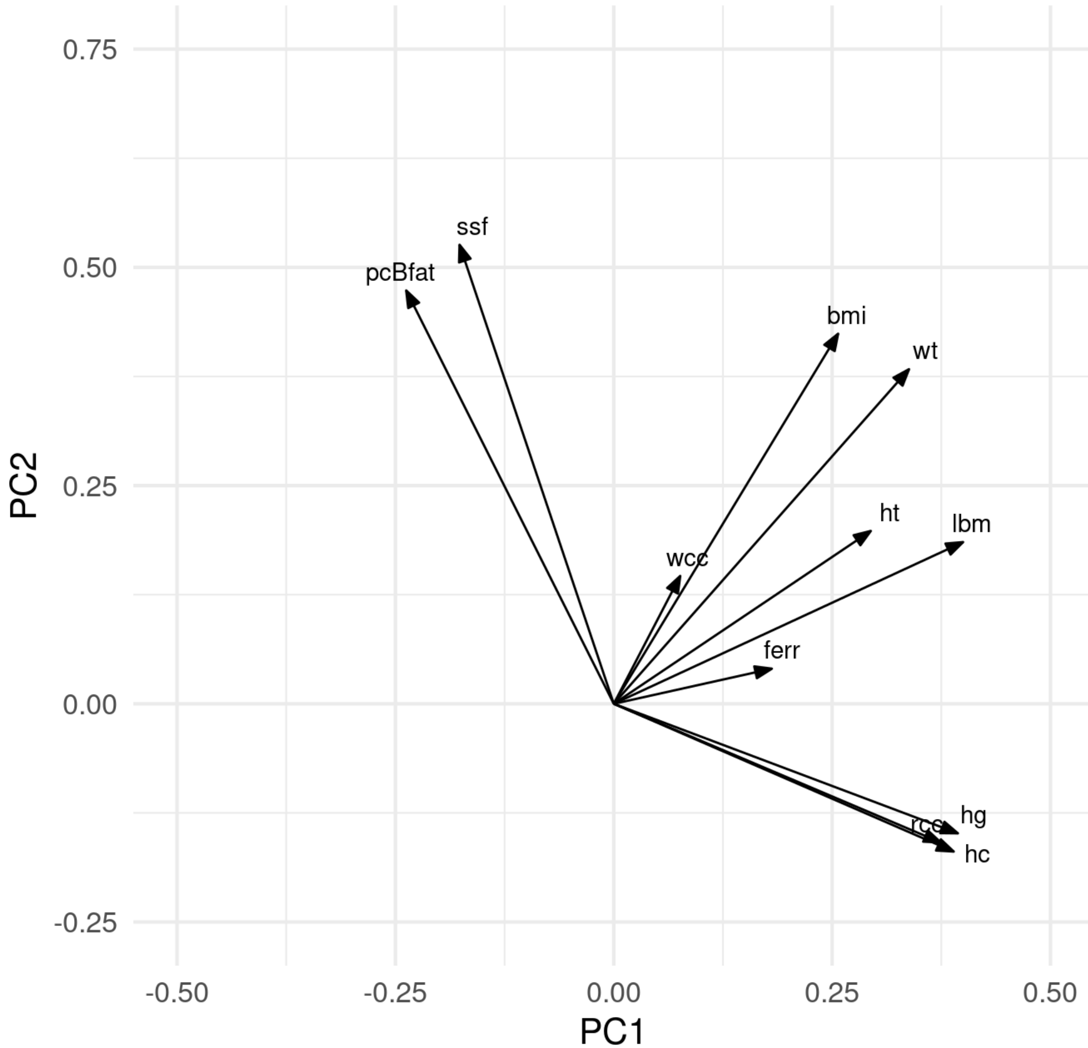
Plotting Embeddings

```r
aus_fit  |>
  augment(aus_athletes) %>%
  ggplot(aes(.fittedPC1, .fittedPC2,color=sex)) +
  geom_point(size = 1.5) +
  scale_color_manual(
    values = c(m = "#D55E00", f = "#0072B2")
  ) +
  theme_minimal(base_size=16) +
  labs(title="PCA Coordinates for Australian Athletes",
       x = "PC 1", y= "PC 2",subtitle = "Color shows sex")
```

Plotting Embeddings



Drawing Clusters



Drawing Clusters

```r
arrow_style <- arrow(
  angle = 20, length = grid::unit(8, "pt"),
  ends = "first", type = "closed"
)
aus_fit %>%
  tidy(matrix = "rotation")  |>
  pivot_wider(
    names_from = "PC", values_from = "value",
    names_prefix = "PC"
  )  |>
  ggplot(aes(PC1, PC2)) +
  geom_segment(
    xend = 0, yend = 0,
    arrow = arrow_style
  ) +
  geom_text_repel(aes(label = column), hjust = 1,vjust=1) +
  xlim(-0.5, 0.5) + ylim(-0.25, 0.75) +
  coord_fixed() +
  theme_minimal(base_size =16)
```

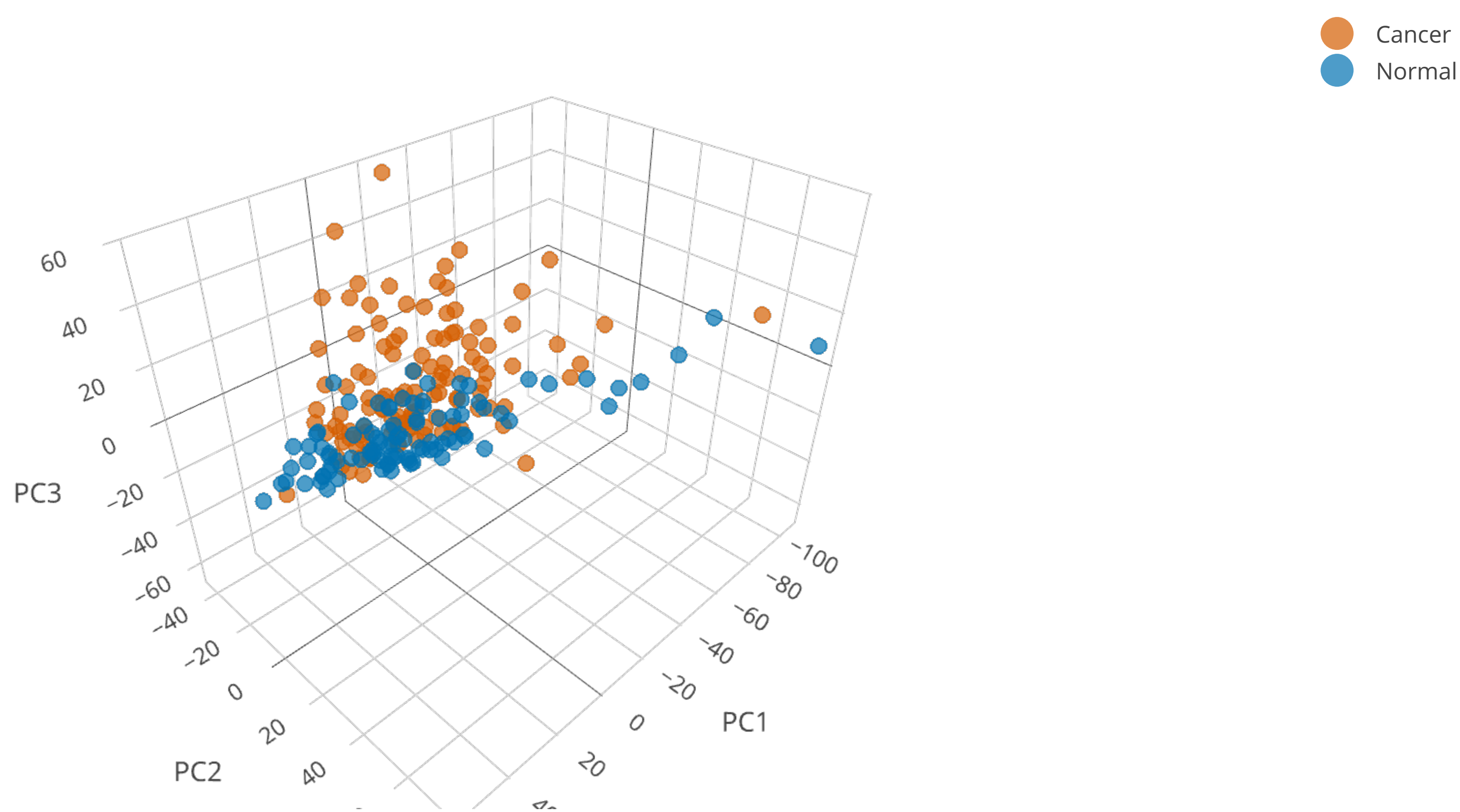
3D Plot Using Plotly

```
library(plotly)

fig = pca_fit |>
 augment(oc_data)  |>
  bind_cols(oc_group) |> plot_ly(
    x=~.fittedPC1, y=~.fittedPC2, z=~.fittedPC3, color = ~Cancer, colors =
fig <- fig |>  add_markers()
fig <- fig |> layout(scene = list(xaxis = list(title = "PC1"),
                     yaxis = list(title = 'PC2'),
                     zaxis = list(title = 'PC3'))
                     )

fig
```

3D Plot Using Plotly



Downsides of PCA

- PCA learns a linear representation
  - Tons of other methods exist for nonlinear structures

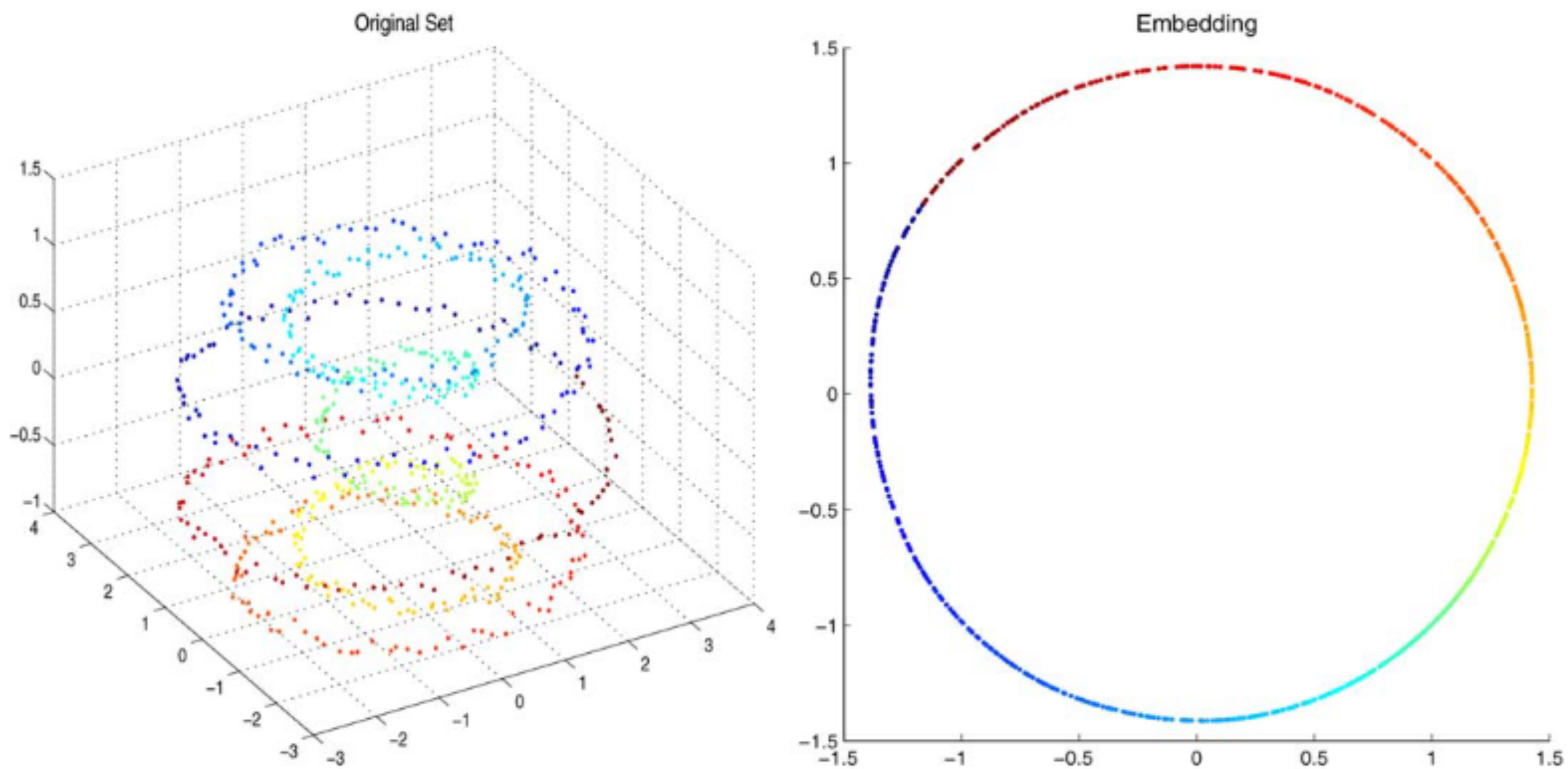
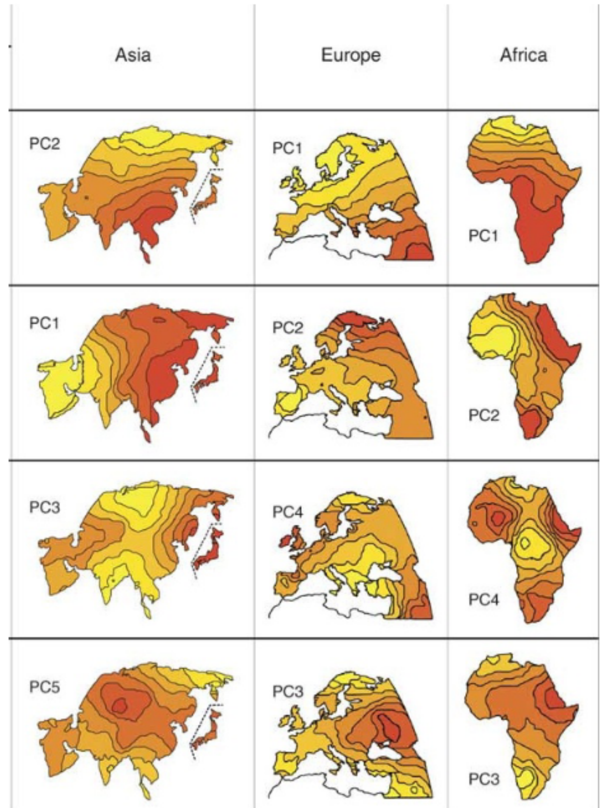


Fig. 6. Left: a noisy helix. Right: the curve is embedded as a perfect circle.

Downsides of PCA

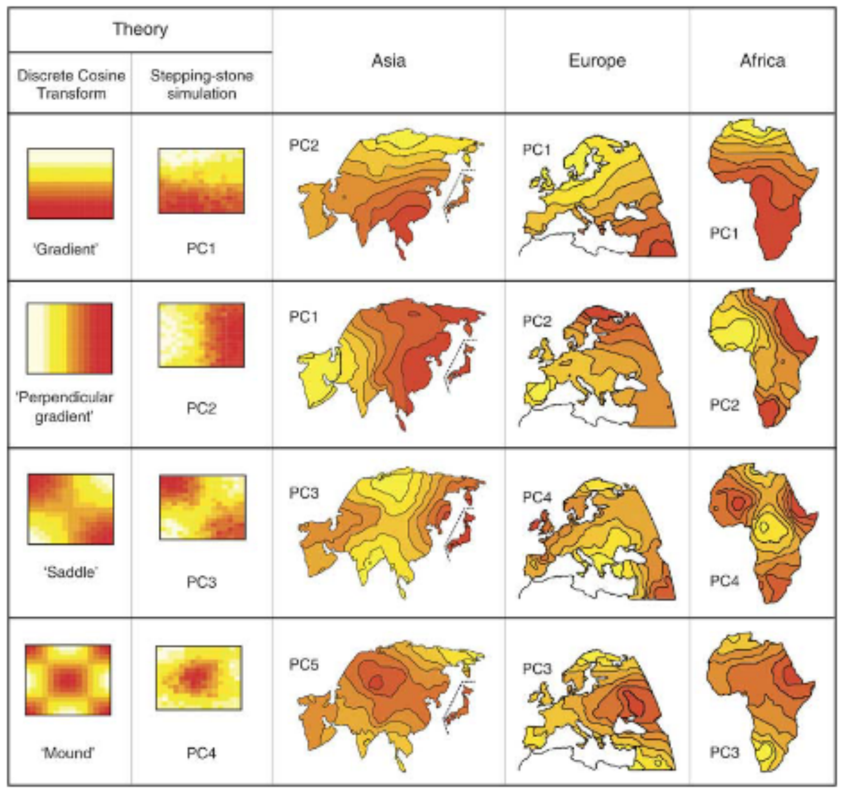
- Prone to overinterpretation



Shalizi/November

Downsides of PCA

- Prone to overinterpretation



Shalizi/November

Downsides of PCA

- Sensitive to outliers
  - If your measurement process is not robust….
  - Using PCA could be very misleading!
- There is an algorithm called robust PCA which is great these cases