Análisis de datos con Stat y Caret

Emmanuel N. Millán
FCEN 2017

Introducción

Caret significa **C**lassification **A**nd **RE**gression **T**raining.

El paquete caret contiene un conjunto de herramientas para construir modelos con Machine Learning o Aprendizaje Automático en R.

Herramientas que provee:

- Preparación de datos, imputación, centrado/ajuste de datos, remoción de predictores correlacionados, reducción de falta de simetría.
- División de datos
- Evaluación de modelos
- Selección de variables

Características de caret

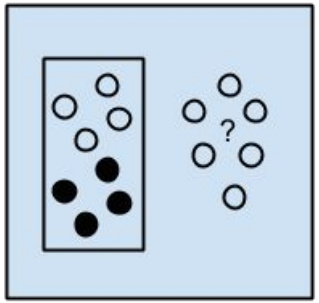
- La mayoría de las herramientas de Machine Learning (ML) en R se utilizan de distinta forma y son difíciles de integrar entre sí.

- caret soluciona este problema, provee una interfaz común para acceder a un número importante de modelos y métodos de machine learning (ML).

- Se puede acceder a métodos de ML como: regresión lineal, redes neuronales, SVM (Support Vector Machine), randomForest, k nearest neighbors, etc.

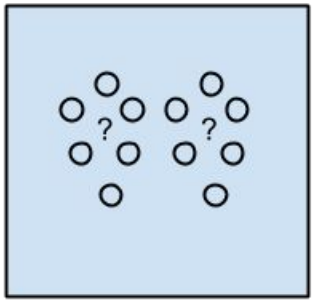
Algoritmos de ML: estilos de aprendizaje

Aprendizaje supervisado



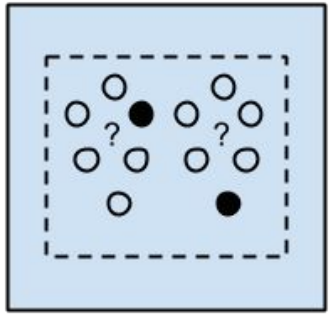
Supervised Learning Algorithms

Aprendizaje no supervisado



Unsupervised Learning Algorithms

Aprendizaje semi-supervisado

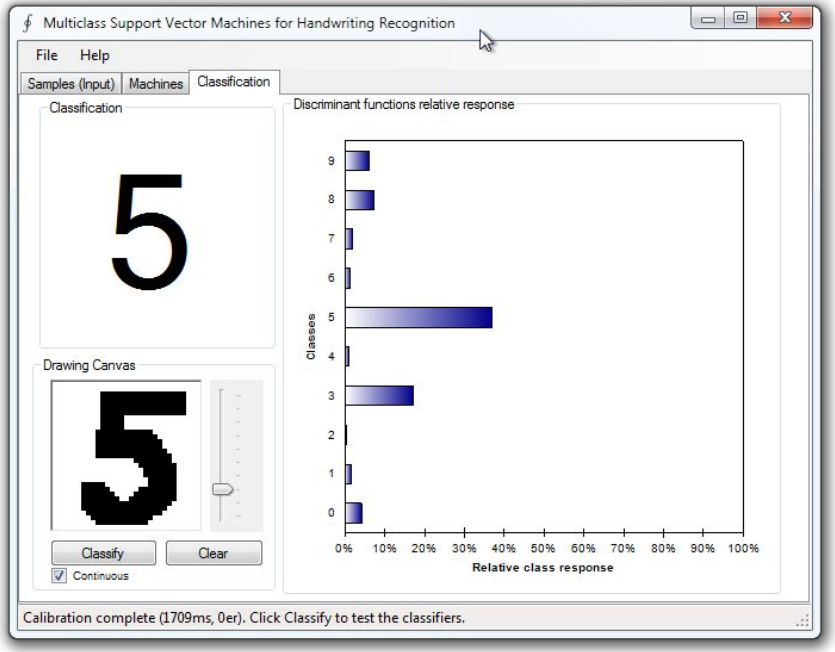


Semi-supervised Learning Algorithms

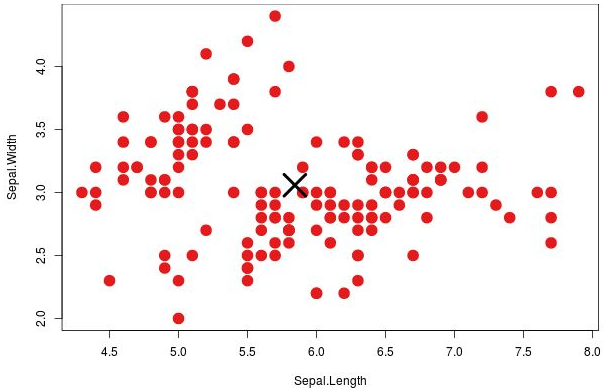
Aprendizaje supervisado

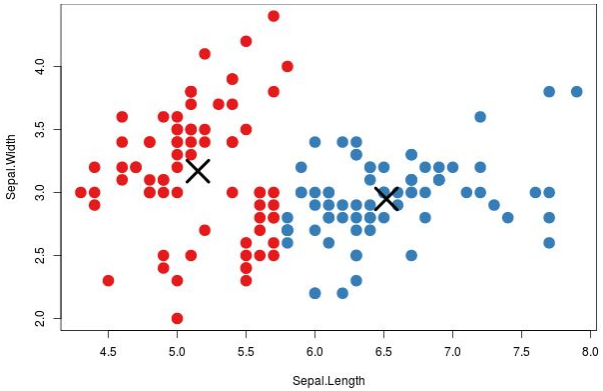
Entrenamiento

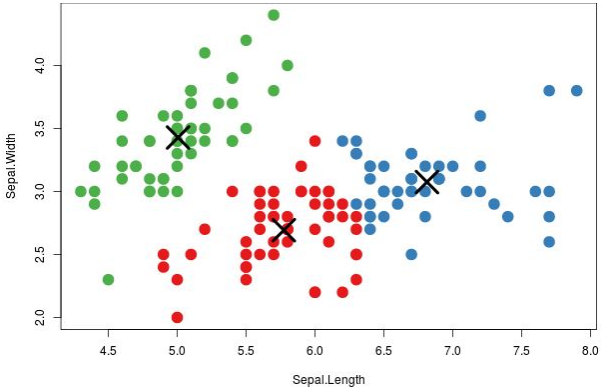
Clasificación

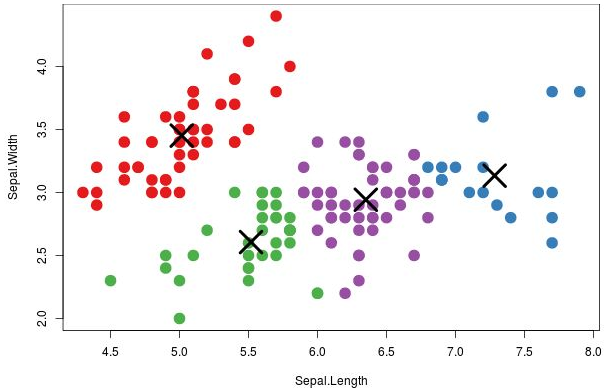


Aprendizaje no supervisado









Set de datos

- Se utilizo un set de datos descargado de:
  http://www.basketball-reference.com/players/g/ginobma01.html
- Este set de datos tiene las estadísticas de juego por año de Emanuel Ginobili en la NBA.
- Se graficó la relación de Minutos Jugados (MP) y Puntos Anotados (PTS).
- El set de datos completo se puede ver en el siguiente slide.

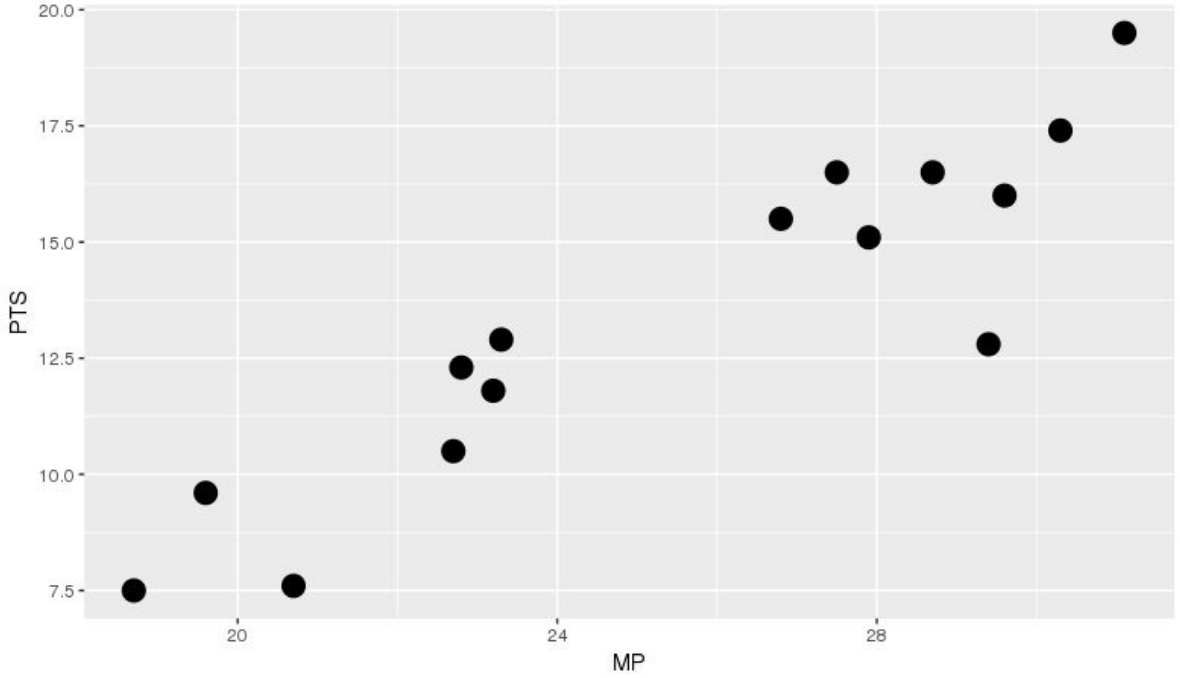
Manu Ginobili estadísticas por juego anuales

Per Game Share & more ▾ Glossary

Season	Age	Tm	Lg	Pos	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
2002-03	25	SAS	NBA	SG	69	5	20.7	2.5	5.8	.438	0.7	2.1	.345	1.8	3.6	.494	.503	1.8	2.5	.737	0.7	1.7	2.3	2.0	1.4	0.2	1.4	2.5	7.6
2003-04	26	SAS	NBA	SG	77	38	29.4	4.3	10.2	.418	1.1	3.2	.359	3.1	7.1	.445	.474	3.1	3.9	.802	1.1	3.4	4.5	3.8	1.8	0.2	2.1	2.4	12.8
2004-05 ★	27	SAS	NBA	SG	74	74	29.6	5.0	10.5	.471	1.3	3.5	.376	3.6	7.1	.517	.533	4.8	6.0	.803	1.0	3.4	4.4	3.9	1.6	0.4	2.3	2.6	16.0
2005-06	28	SAS	NBA	SG	65	56	27.9	4.8	10.3	.462	1.3	3.3	.382	3.5	7.0	.500	.524	4.3	5.5	.778	0.6	2.9	3.5	3.6	1.6	0.4	1.9	2.4	15.1
2006-07	29	SAS	NBA	SG	75	36	27.5	5.3	11.4	.464	1.7	4.3	.396	3.6	7.1	.505	.539	4.3	5.0	.860	0.8	3.6	4.4	3.5	1.5	0.4	2.1	2.1	16.5
2007-08	30	SAS	NBA	SG	74	23	31.1	6.1	13.3	.460	2.1	5.3	.401	4.0	8.0	.499	.540	5.1	6.0	.860	0.9	3.9	4.8	4.5	1.5	0.4	2.7	2.3	19.5
2008-09	31	SAS	NBA	SG	44	7	26.8	5.1	11.2	.454	1.6	4.8	.330	3.5	6.4	.546	.524	3.8	4.3	.884	0.5	4.0	4.5	3.6	1.5	0.4	2.0	2.0	15.5
2009-10	32	SAS	NBA	SG	75	21	28.7	5.3	12.0	.441	1.8	4.7	.377	3.5	7.4	.481	.514	4.1	4.7	.870	0.9	2.9	3.8	4.9	1.4	0.3	2.1	2.1	16.5
2010-11 ★	33	SAS	NBA	SG	80	79	30.3	5.5	12.7	.433	1.9	5.5	.349	3.6	7.2	.497	.509	4.5	5.1	.871	0.5	3.2	3.7	4.9	1.5	0.4	2.2	2.0	17.4
2011-12	34	SAS	NBA	SG	34	7	23.3	4.4	8.4	.526	1.5	3.7	.413	2.9	4.7	.616	.618	2.6	3.0	.871	0.5	2.9	3.4	4.4	0.7	0.4	1.9	1.6	12.9
2012-13	35	SAS	NBA	SG	60	0	23.2	3.8	9.0	.425	1.4	3.9	.353	2.4	5.1	.480	.502	2.7	3.4	.796	0.5	2.9	3.4	4.6	1.3	0.2	2.2	1.9	11.8
2013-14	36	SAS	NBA	SG	68	3	22.8	4.3	9.2	.469	1.3	3.8	.349	3.0	5.4	.553	.541	2.4	2.8	.851	0.4	2.5	3.0	4.3	1.0	0.3	2.0	1.9	12.3
2014-15	37	SAS	NBA	SG	70	0	22.7	3.6	8.4	.426	1.3	3.7	.345	2.3	4.7	.489	.502	2.1	2.9	.721	0.4	2.6	3.0	4.2	1.0	0.3	2.2	2.0	10.5
2015-16	38	SAS	NBA	SG	58	0	19.6	3.4	7.5	.453	1.2	3.1	.391	2.2	4.4	.496	.533	1.6	1.9	.813	0.4	2.1	2.5	3.1	1.1	0.2	1.7	1.7	9.6
2016-17	39	SAS	NBA	SG	69	0	18.7	2.5	6.4	.390	1.3	3.3	.392	1.2	3.1	.387	.491	1.2	1.6	.804	0.4	1.9	2.3	2.7	1.2	0.2	1.4	1.7	7.5
Career			**NBA**		**992**	**349**	**25.8**	**4.4**	**9.9**	**.447**	**1.4**	**3.9**	**.370**	**3.0**	**6.0**	**.497**	**.520**	**3.3**	**4.0**	**.826**	**0.7**	**2.9**	**3.6**	**3.9**	**1.4**	**0.3**	**2.0**	**2.1**	**13.6**

scatter plot: geom_point()

```
ggplot(data = manu, aes(x = MP, y = PTS)) + geom_point(size=5)
```



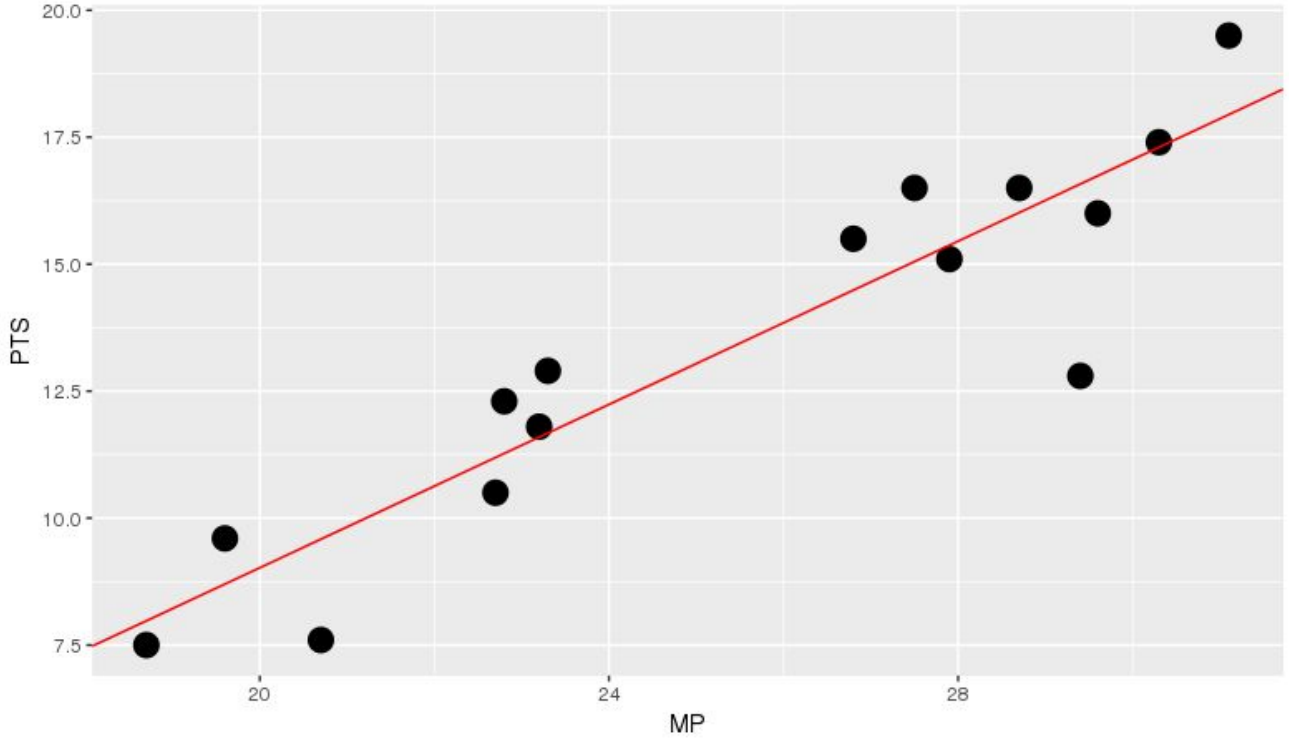
función train() de caret

```
model.manu_caret <- train(PTS ~ MP, data=manu, method="lm")
coef.icept <- coef(model.manu_caret$finalModel)[1]
coef.slope <- coef(model.manu_caret$finalModel)[2]
pl <- ggplot(data = manu, aes(x = MP, y = PTS))
pl <- pl + geom_point(size=5)
pl <- pl + geom_abline(slope = coef.slope, intercept = coef.icept, color="red")
pl
```

Parámetros de la función train():

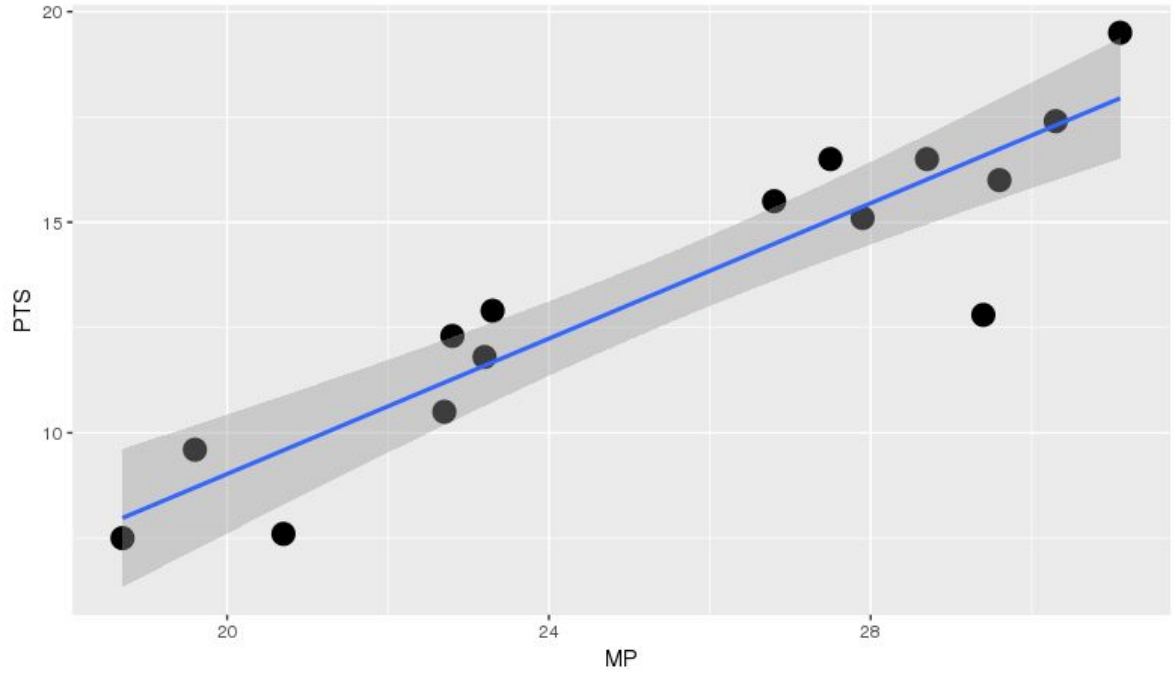
- Dataset con el que se va a trabajar
- La variable que se desea predecir (PTS)
- La variable de entrada (MP)
- El método de ML que se desea utilizar (lm o linear regression)
- La sintaxis PTS ~ MP le dice a caret “Quiero predecir PTS utilizando como base una sola variable, MP”

caret con método “lm” o linear regression



geom_smooth con method=”lm”

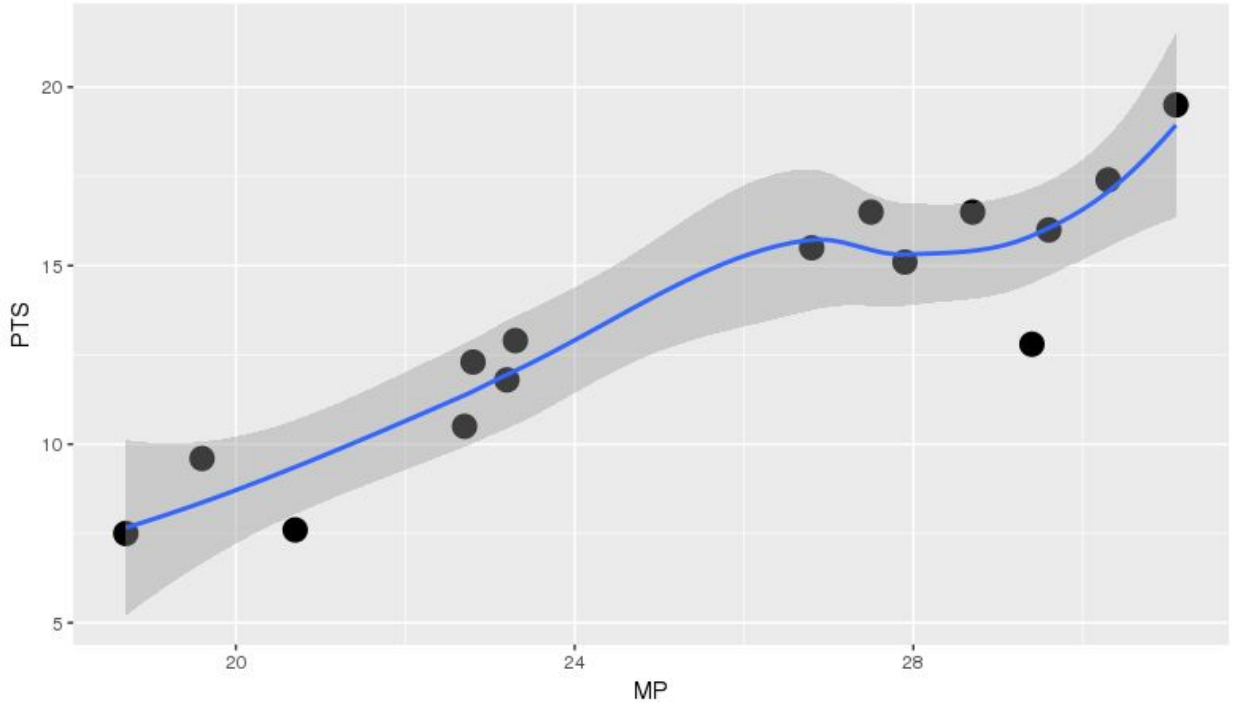
```
ggplot(data = manu, aes(x = MP, y = PTS)) + geom_point(size=5) + geom_smooth(method="lm")
```



geom_smooth con method=”loess”

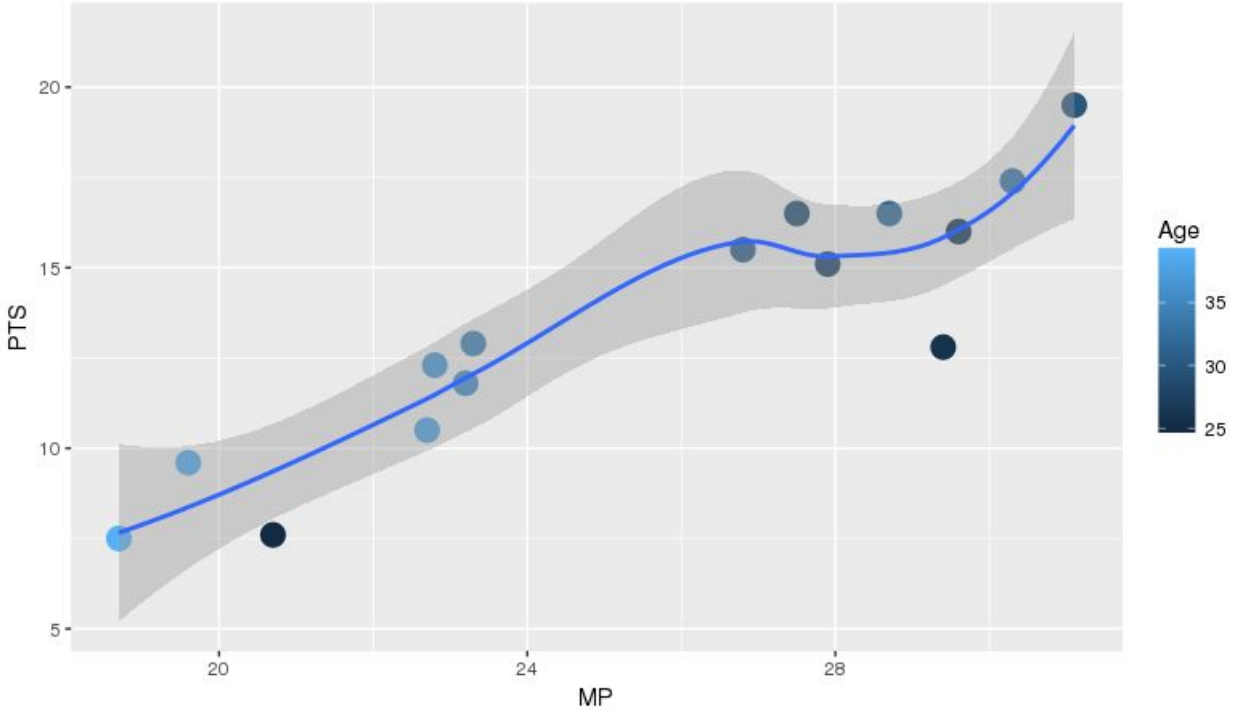
```
ggplot(data = manu, aes(x = MP, y = PTS)) + geom_point(size=5) + geom_smooth(method="loess")
```

loess:
Local Polynomial
Regression Fitting

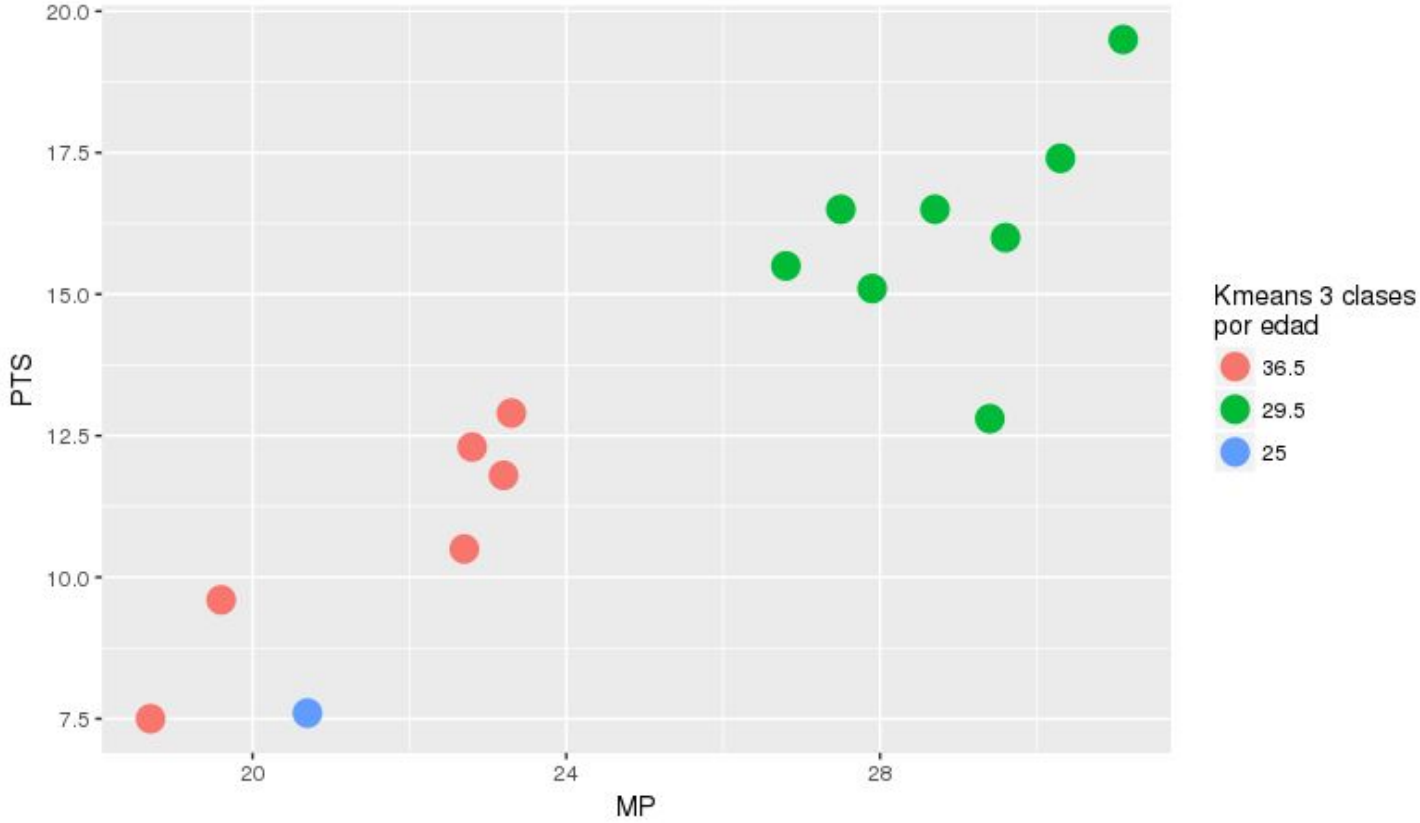


MP vs PTS y Edad por color

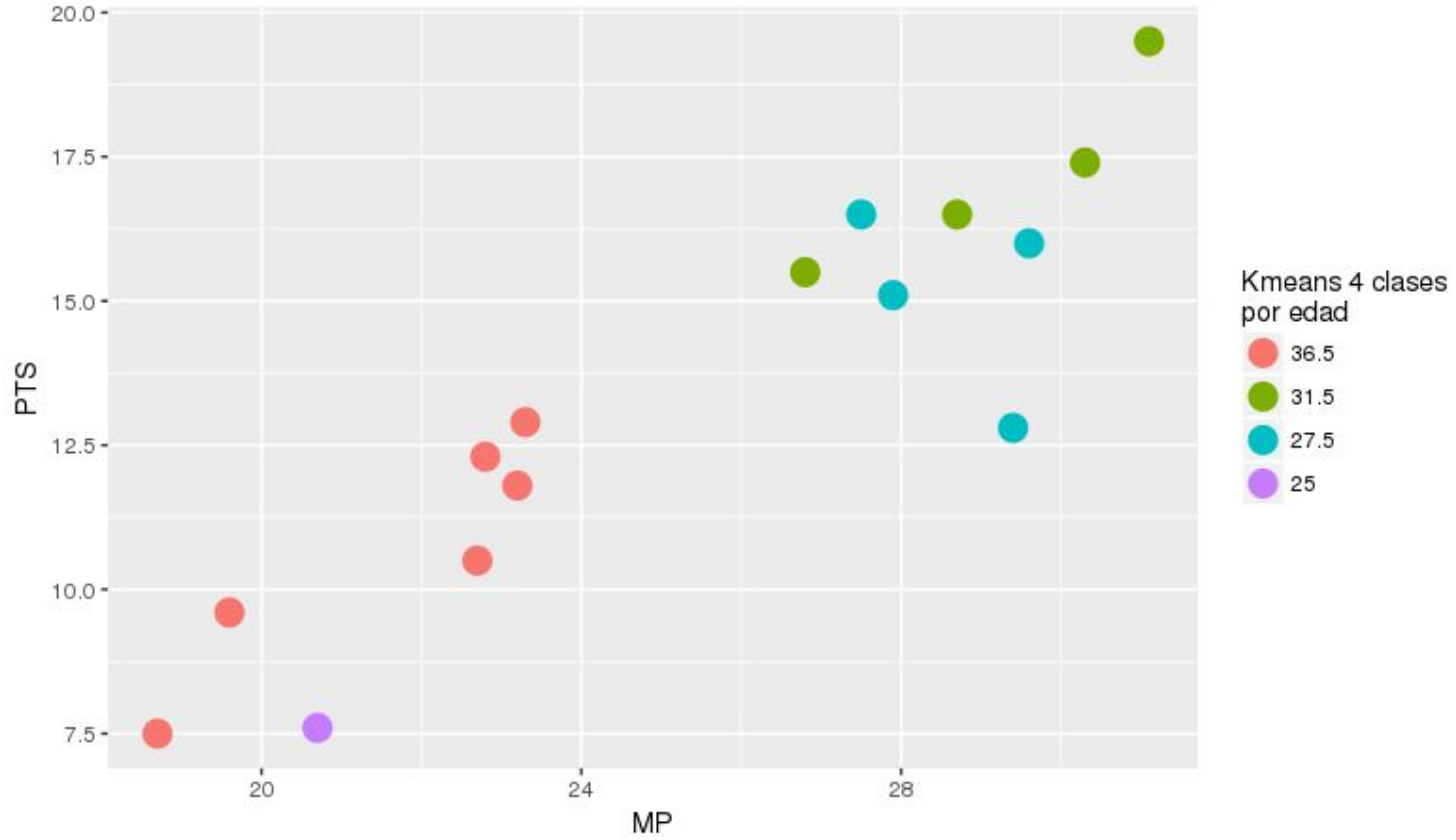
```
ggplot(data = manu, aes(x = MP, y = PTS, color=Age)) + geom_point(size=5) + geom_smooth(method="loess")
```



Análisis con kmeans, 3 clases



Análisis con kmeans, 4 clases



Código Kmeans

```
manu_d <- dplyr::select(data=manu, Age, MP)

manuCluster <- kmeans(manu_d, 4, nstart=20)

manuCluster$cluster <- as.factor(manuCluster$cluster)

pl <- ggplot(manu, aes(MP, PTS, color=manuCluster$cluster))

pl <- pl  +  geom_point(size=5)

pl <- pl  + scale_color_discrete(name="Kmeans 4 clases\npor edad",
            breaks=c("1","2","3","4"),
            labels=c("36.5","31.5","27.5","25"))

pl
```

Referencias

- http://sharpsightlabs.com/blog/quick-introduction-machine-learning-r-caret/
- http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/
- https://datascienceplus.com/k-means-clustering-in-r/
- http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/
- https://shiny.rstudio.com/gallery/kmeans-example.html