In [1]:
```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA
from sklearn.metrics import mean_squared_error, r2_score


from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, plot_con
fusion_matrix

from sklearn.model_selection import train_test_split # simple TT split
cv
from sklearn.model_selection import cross_val_score # cross validation
metrics
from sklearn.model_selection import cross_val_predict # cross validati
on metrics

from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
import scipy.cluster.hierarchy as sch
from matplotlib import pyplot as plt

from sklearn.metrics import silhouette_score

%matplotlib inline
```

Abstract: Before looking at the data below it is important that we clarify some of the variables being used. Active Spin: Spin on the baseball that contributes to movement, if the number is 100, then that means the pitch is thrown with pure backspin or topspin. Fastball: Most common pitch thrown by pitchers, tends to go straight but can have some variation of movement. Cutter: A pitch that is similar in speed to the fastball but breaks to the pitcher's glove-hand side. Changeup: A pitch thrown that is supposed to look like a fastball but then drop towards the end. Used as a deception pitch and is slower than a fastball. Slider: A pitch thrown that moves similarly to a cutter with more lateral break, (usually) more horizontal break, and less velocity. Curveball: A pitch thrown that induces top spin on the ball causing it to break down and to the pitcher's glove side with the velocity being much slower than a fastball. There are multiple varieties of a curveball and is categorized by how the ball breaks (A 12-6 Curveball breaks pretty much up and down (hence 12 o'clock to 6 o'clock), a sweeping curve breaks from 2 o'clock to 7 or 8 o'clock. K/9: Based on the pitcher's sample size, an estimate of how many batters they would strike out per 9 innings. Average K/9 is 7.7 (20% of batters in 9 innings). ERA: A statistic used to measure a pitcher's effectiveness, obtained by calculating the average number of earned runs scored against the pitcher in every nine innings pitched (average 4.2). BB/9: Based on the pitcher's sample size, an estimate of how many batter they would walk per 9 innings. Average BB/9 is 2.9 (7.7% of batters in 9 innings). The data was collected from the official MLB site where we took all qualifying right handed pitchers. Values used in the question were based off of averages of qualifying pitchers. Other values that were used derived from league averages stated above.

In [4]:
```
baseball = pd.read_csv("https://raw.githubusercontent.com/BShimabuku/FinalFerda/main/Active%20Spin%20%20-%20Sheet2%20(1).csv")
baseball.head()
```

Out[4]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change	ac
0	Mahle, Tyler	CIN	R	99.9	35.810811	88.1	
1	Cole, Gerrit	NYY	R	99.9	35.810811	99.0	
2	May, Dustin	LAD	R	97.7	30.500000	83.7	
3	Urquidy, Jose	HOU	R	96.7	35.810811	92.8	
4	Gonsolin, Tony	LAD	R	96.6	35.810811	90.1	

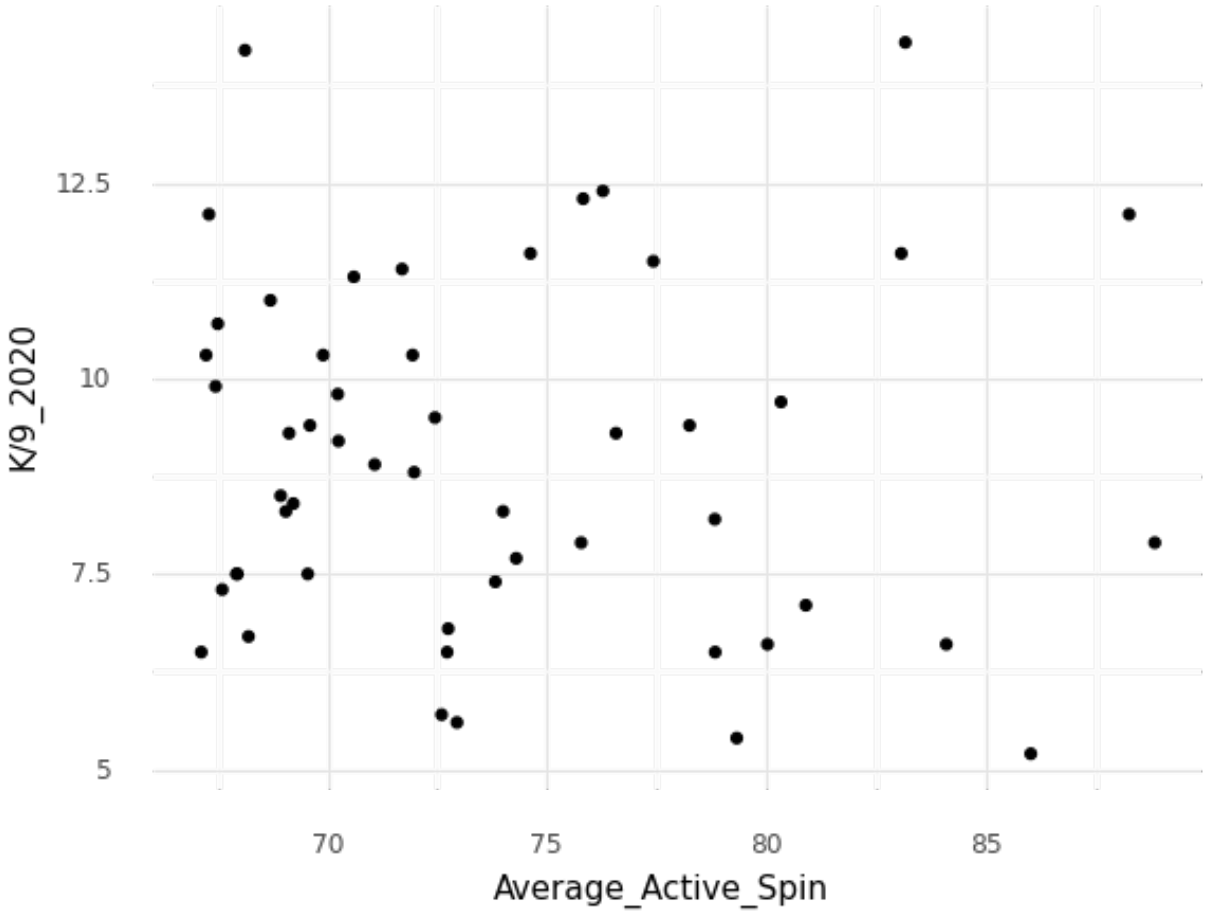
Q1: What groups are formed when looking at pitchers with an average active spin above 67% and an above average K/9? (EM)

In [ ]:
```
HActSpin = baseball.loc[baseball.Average_Active_Spin >= 67]
HActSpin.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change	ac
0	Mahle, Tyler	CIN	R	99.9	35.810811	88.1	
1	Cole, Gerrit	NYY	R	99.9	35.810811	99.0	
3	Urquidy, Jose	HOU	R	96.7	35.810811	92.8	
4	Gonsolin, Tony	LAD	R	96.6	35.810811	90.1	
5	Cobb, Alex	BAL	R	96.5	35.810811	73.9	

In [ ]:
```
(ggplot(HActSpin, aes("Average_Active_Spin", "K/9_2020")) + geom_point
() + theme_minimal())
```



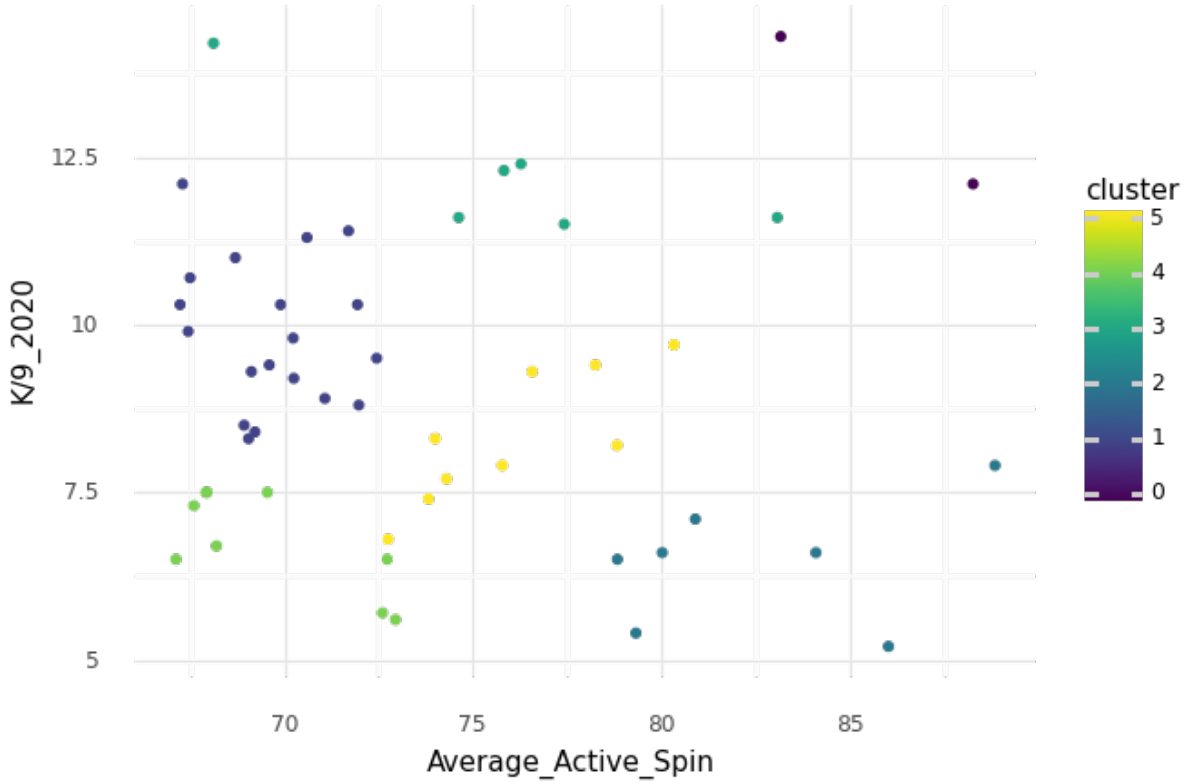
Out[ ]: <ggplot: (8754521669566)>

This graph shows the relationship between the variables K/9 and average active spin above 67%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["Average_Active_Spin", "K/9_2020"]

X = HActSpin[features]
z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)
EM.fit(X)

cluster = EM.predict(X)

X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]:
```
0.6398049575595575
```

In [ ]:
```
(ggplot(HActSpin, aes("Average_Active_Spin", "K/9_2020", color = "clus
ter"))+ geom_point() + theme_minimal())
```



Out[ ]: <ggplot: (-9223363273905395922)>

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph we can see that the clusters formed on the left side of the graph show that pitchers with an average active spin around 70 have around league average to above league average K/9 values. Clusters in the middle of the graph show that pitchers with an average active spin around 75-80 also have around league average to above league average K/9 values. Finally, clusters on the right side of the graph show that pitchers with an average active spin around 80-85 have either well above league average or below league average K/9 values.

Answer to Question: When looking at pitchers with an average active spin above 67% and an above average K/9, some groups formed even though there was a lot of variation. Although pitchers with higher than average active spin ranged from both below league average and above league average K/9 values, most groups that formed showed that they had around average or above average K/9 values. Our silhouette score came out to be around 0.639. This means that our model performed decently well but there is room for improvement considering we would want our silhouette score to be around 1 which is perfect. There are many other external factors that may have had an effect on our data collected and thus the silhouette score. Overall, the model performed decently at best.

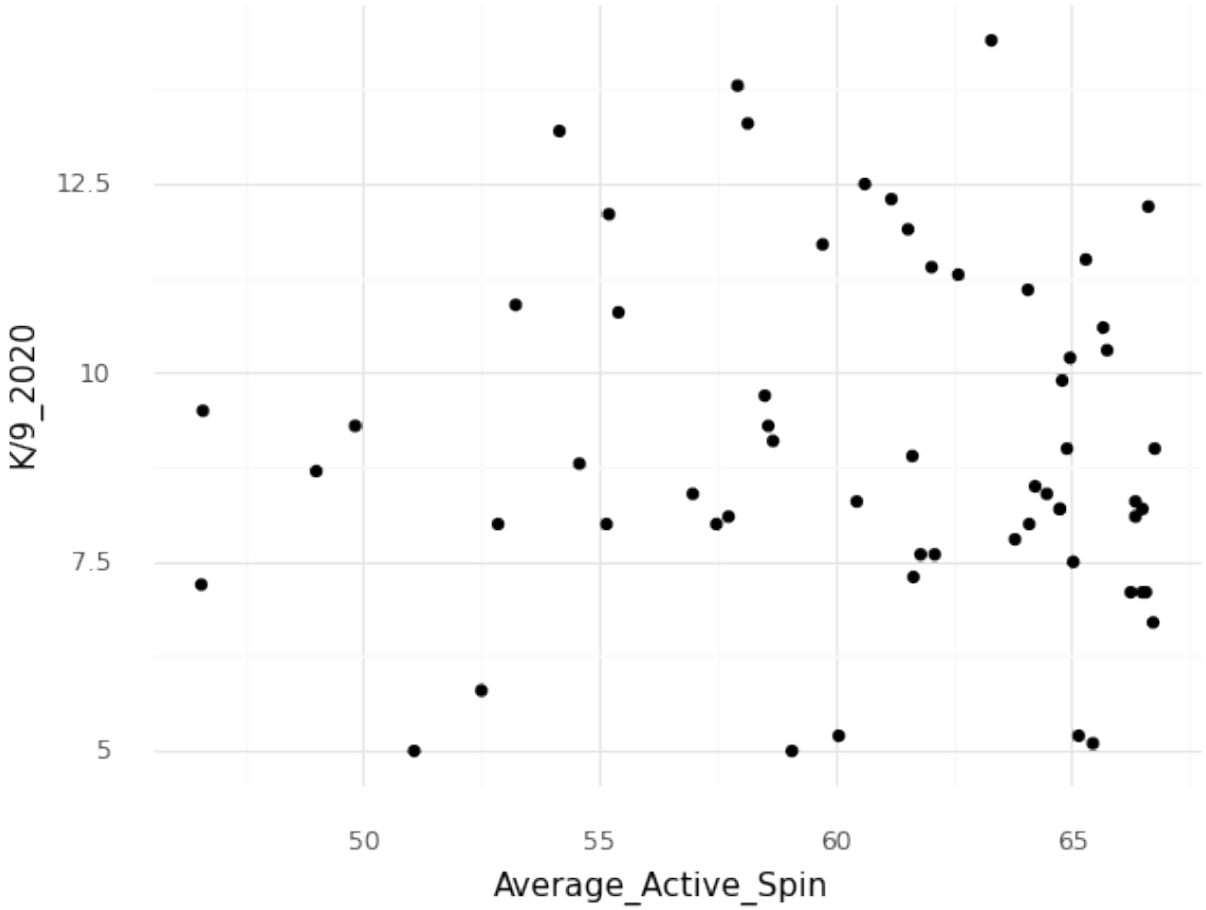
Q2: What groups are formed when looking at pitchers with an average active spin below 67% and a below average K/9?

In [ ]:
```
LActSpin = baseball.loc[baseball.Average_Active_Spin <= 67]
LActSpin.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change
2	May, Dustin	LAD	R	97.7	30.500000	83.7
8	McKenzie, Triston	CLE	R	95.2	35.810811	78.5
14	Quantrill, Cal	CLE	R	93.9	35.810811	77.0
16	Canning, Griffin	LAA	R	93.8	35.810811	94.6
22	Montas, Frankie	OAK	R	92.9	35.810811	72.6

In [ ]:
```
(ggplot(LActSpin, aes("Average_Active_Spin", "K/9_2020")) + geom_point
() + theme_minimal())
```



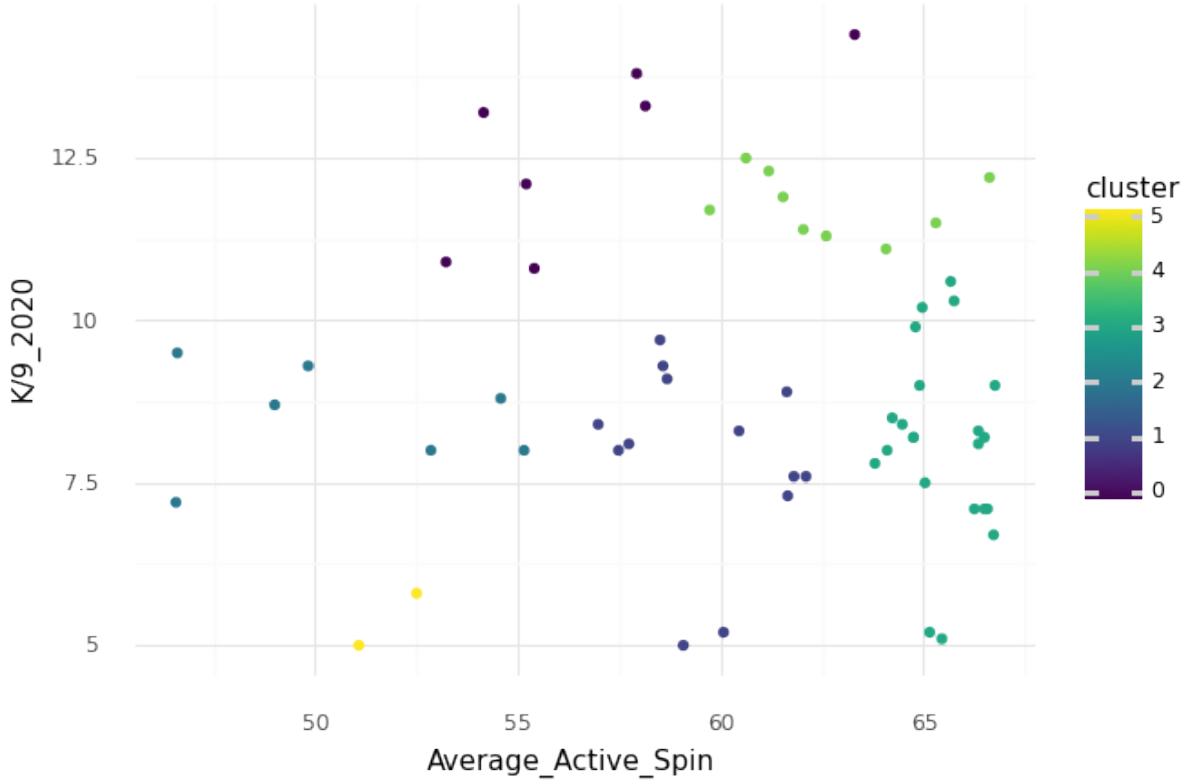
Out[ ]: <ggplot: (-9223363282332845624)>

This graph shows the relationship between the variables K/9 and average active spin below 67%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["Average_Active_Spin", "K/9_2020"]

X = LActSpin[features]
z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)
EM.fit(X)

cluster = EM.predict(X)

X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]: 0.5250385243627367

In [ ]:
```
(ggplot(LActSpin, aes("Average_Active_Spin", "K/9_2020", color = "clus
ter"))+ geom_point() + theme_minimal())
```



Out[ ]: <ggplot: (8762949354651)>

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph we can see that clusters formed on the left side and the middle. These clusters represent pitchers who have around an average active spin of 50 have around league average or below league average K/9 values. There is one cluster with pitchers around 55 average active spin who all have above league average K/9 values. On the right side of the graph it shows pitchers with average active spin around 60-65 to have below league average to above league average K/9 values. However, most of the points on the right hand side clusters are placed as around league average K/9 values.

Answer to Question: Looking at pitchers with an average active spin below 67% and a below average K/9, there was variation but some groups still formed. Pitchers with lower average active spin again ranged from below league average and above league average K/9 values. The groups formed were kind of hard to distinguish and there was some overlapping. This was reflected in our silhouette score of 0.525. As data scientists we aim to achieve a silhouette score of 0.7 or higher, which would mean that our model performed well. We did not achieve this score and it showed in data visualization of our model.

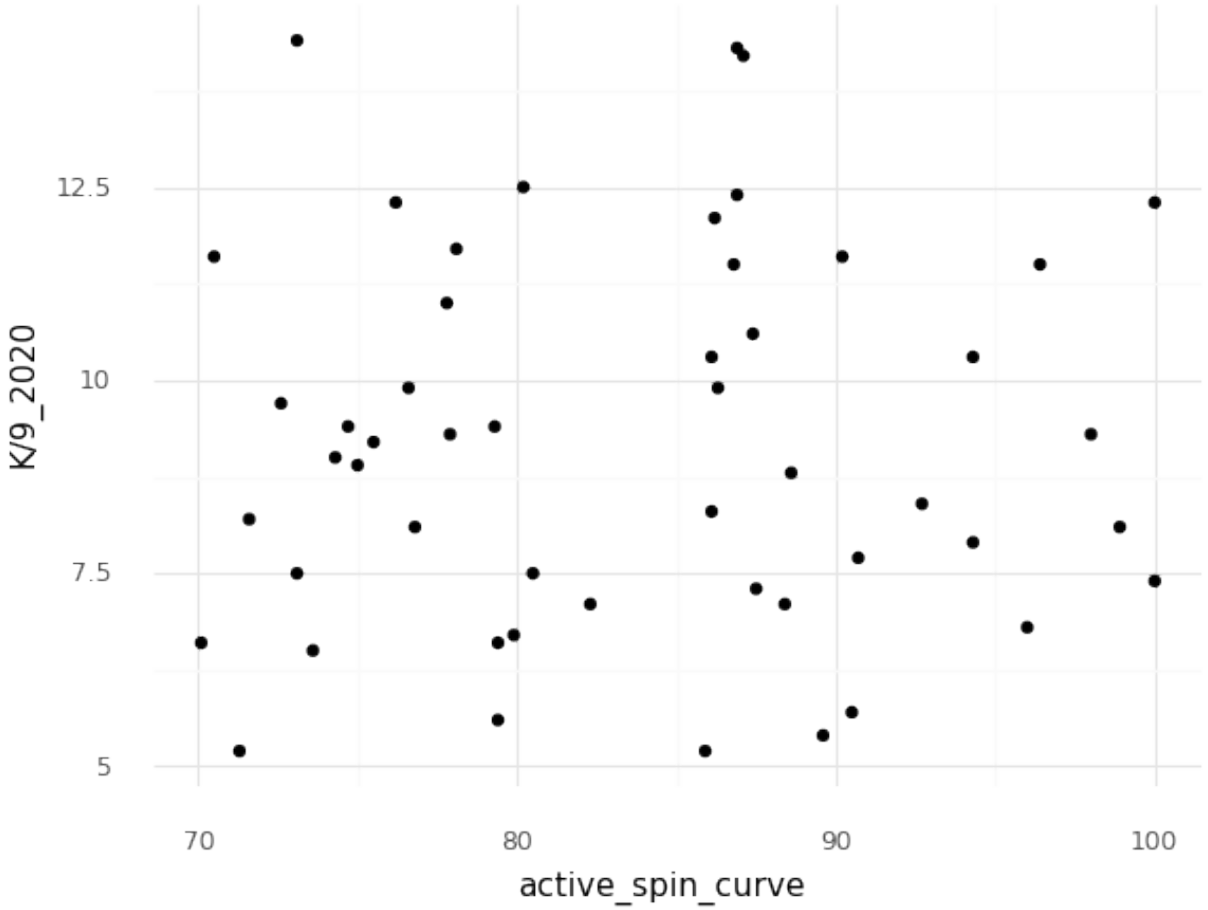
Q3: What groups are formed when looking at pitchers with a curveball above 70% active spin and an above average K/9?

In [ ]:
```
hcurve = baseball.loc[baseball.active_spin_curve >= 70]

hcurve.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change	ac
1	Cole, Gerrit	NYY	R	99.9	35.810811	99.000000	
3	Urquidy, Jose	HOU	R	96.7	35.810811	92.800000	
4	Gonsolin, Tony	LAD	R	96.6	35.810811	90.100000	
6	Ponce de Leon, Daniel	STL	R	95.9	29.800000	92.500000	
7	Bauer, Trevor	CIN	R	95.7	28.300000	87.418182	

In [ ]:
```
(ggplot(hcurve, aes("active_spin_curve", "K/9_2020")) + geom_point() +
theme_minimal())
```



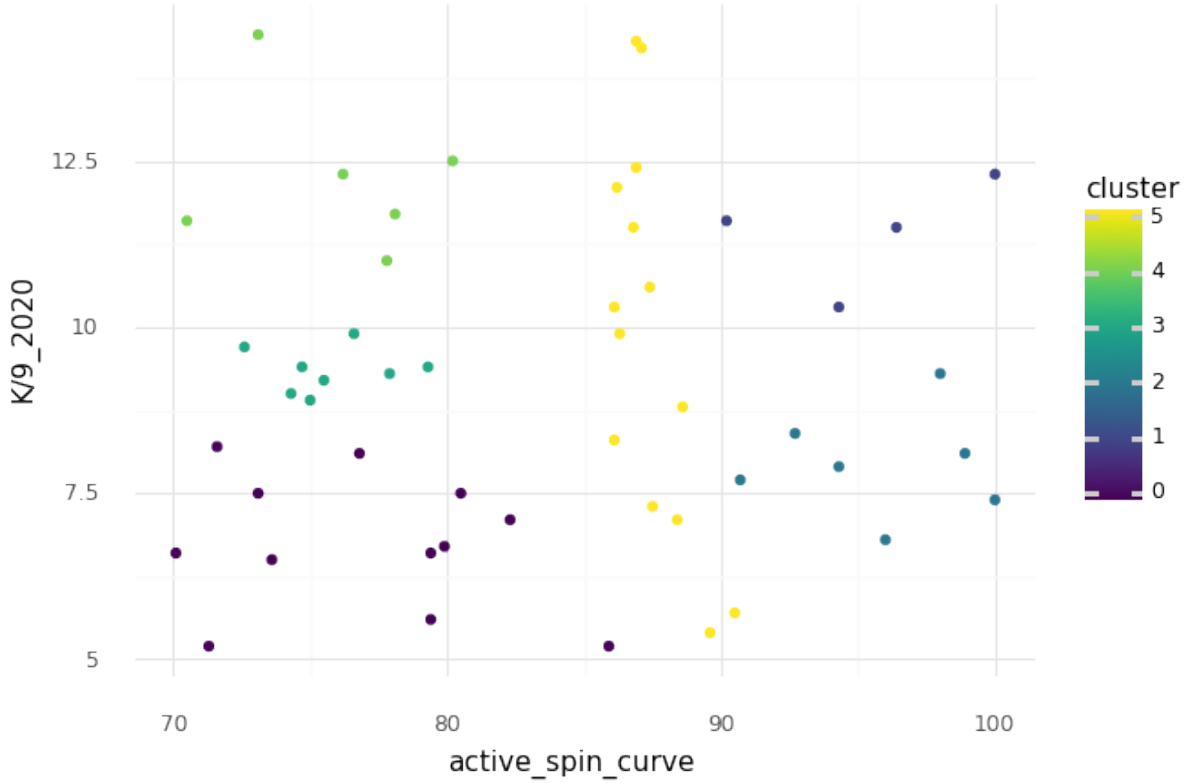
Out[ ]: <ggplot: (-9223363282332952544)>

This graph shows the relationship between the variables K/9 and active spin of a curveball above 70%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["active_spin_curve", "K/9_2020"]

X = hcurve[features]

z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)

EM.fit(X)

cluster = EM.predict(X)
X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]:
```
0.5577610600393192
```

In [ ]:
```
(ggplot(hcurve, aes("active_spin_curve", "K/9_2020", color = "cluster"
))+ geom_point() + theme_minimal())
```



Out[ ]:
```
<ggplot: (-9223632486483701 23)>
```

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph we see clusters formed on the left side show pitchers who have an average active spin on their curveball around 70-80 to have below league average to above league average K/9 values. In the middle the same occurrence happened, as pitchers who have an average active spin on their curveball around 85-90 to have below league average to above league average K/9 values. On the right side where pitchers who have an average active spin on a curve of 90-100 led to an around league average to above league average K/9 value.

Answer to question: When looking at pitchers with an average active spin above 70% on their curveball and an above average K/9 some groups formed despite a lot of variation. Pitchers with slightly above 70% average active spin on their curveball had a ranging cluster from below league average to above league average K/9 values. Pitchers with an average active spin near 100% showed that they had around league average or above league average K/9 values. Although these groups were established, there was a fair amount of overlapping and it shows since our silhouette score came out to be around 0.558. We would like to have had our silhouette score around 0.7 which would have indicated that our model performed well.

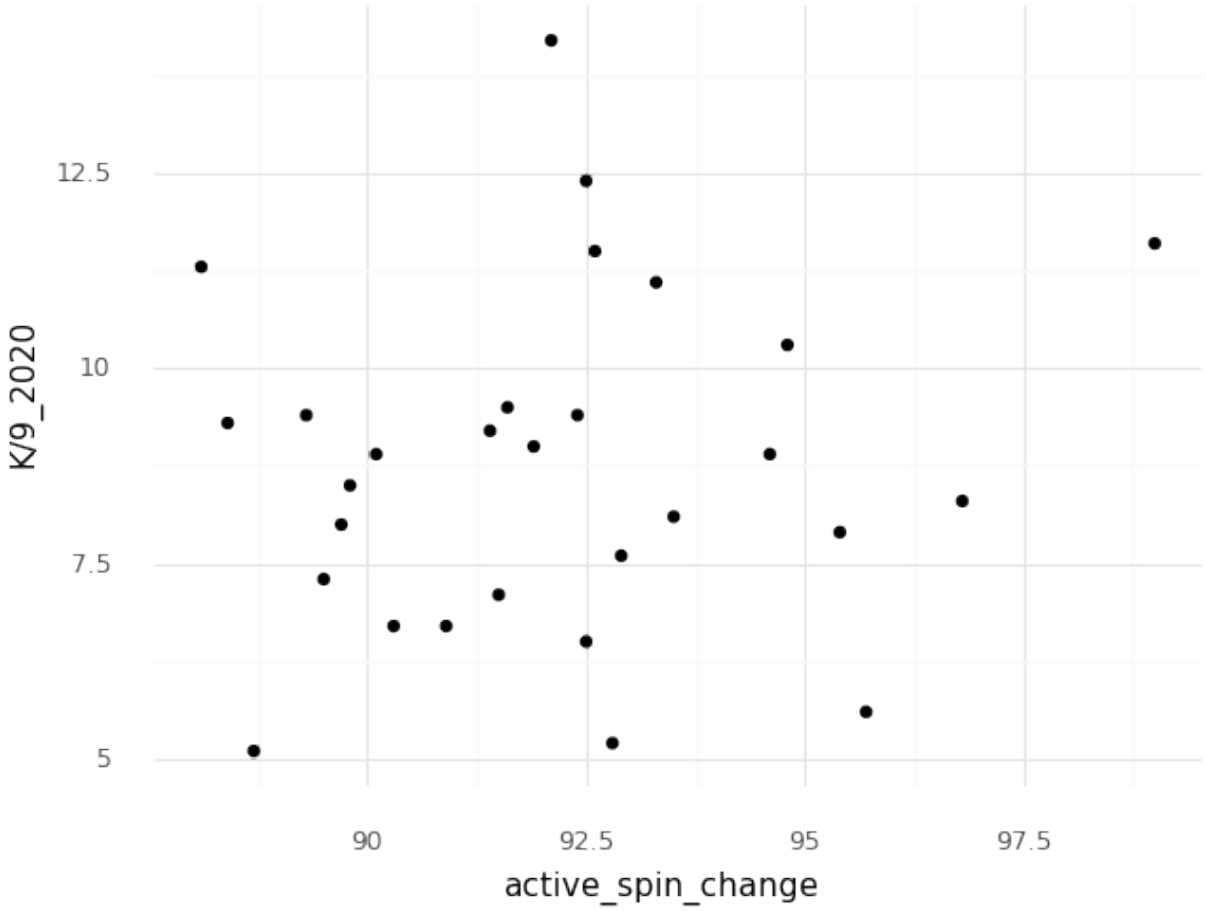
Q4: What groups are formed when looking at pitchers with a changeup above 80% active spin and an above average K/9?

In [ ]:
```
hchange = baseball.loc[baseball.active_spin_change >= 88]

hchange.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change	ac
0	Mahle, Tyler	CIN	R	99.9	35.810811	88.1	
1	Cole, Gerrit	NYY	R	99.9	35.810811	99.0	
3	Urquidy, Jose	HOU	R	96.7	35.810811	92.8	
4	Gonsolin, Tony	LAD	R	96.6	35.810811	90.1	
6	Ponce de Leon, Daniel	STL	R	95.9	29.800000	92.5	

In [ ]:
```
(ggplot(hchange, aes("active_spin_change", "K/9_2020")) + geom_point()
+ theme_minimal())
```



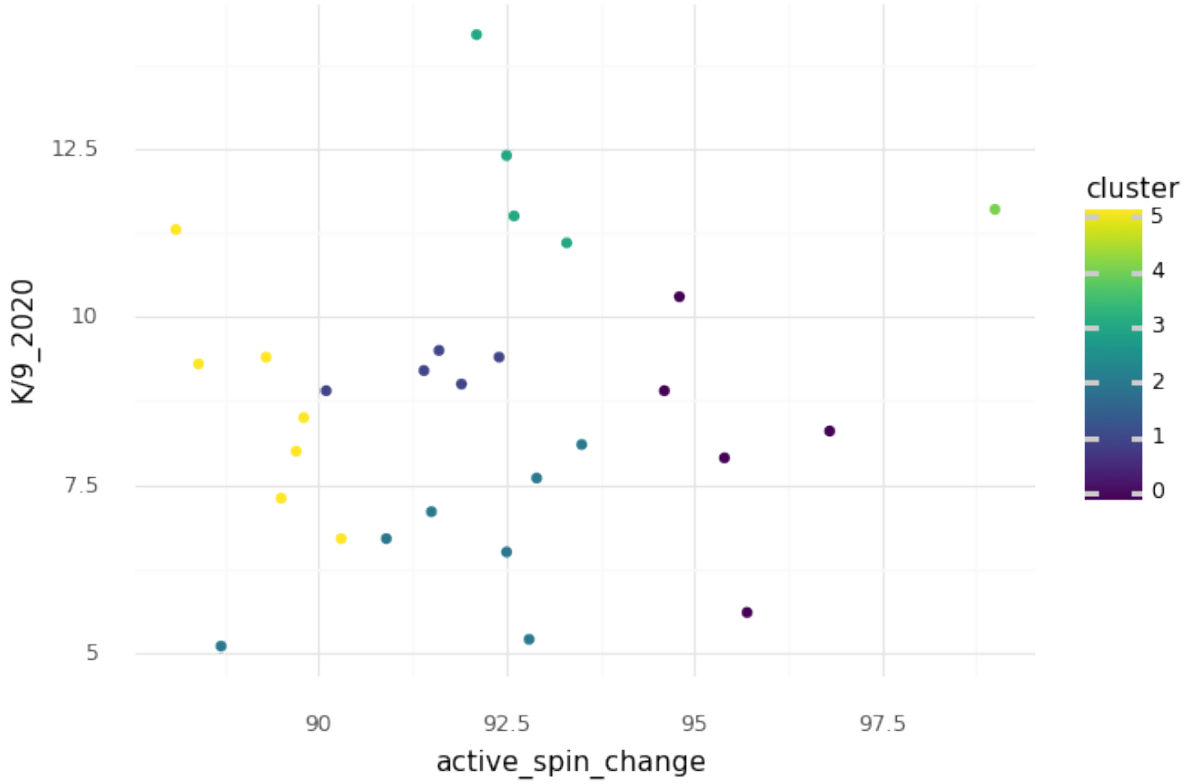
Out[ ]: <ggplot: (8788203749668)>

This graph shows the relationship between the variables K/9 and average active spin of a changeup above 88%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["active_spin_change", "K/9_2020"]

X = hchange[features]

z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)

EM.fit(X)

cluster = EM.predict(X)
X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]:
```
0.5331161529704738
```

In [ ]:
```
(ggplot(hchange, aes("active_spin_change", "K/9_2020", color = "cluste
r"))+ geom_point() + theme_minimal())
```



Out[ ]:
```
<ggplot: (-9223632486483988 69)>
```

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph we see clusters formed on the left side show pitchers who have an average active spin on their changeup around 88-90 tend to have below league average to around league average K/9 values. The middle cluster indicates pitchers who have an active spin on a changeup around 92.5 range from below league average to above league average K/9 values. Of pitchers who have an average active spin on their changeup around 95-100 the right side cluster indicates that they too have around league average to above league average K/9 values with 1 data point being below league average K/9 value.

Answer to Question: Looking at pitchers with an average active spin above 88% on their changeup and an above average K/9 value, groups did form though there was variation. All groups showed variation that had ranges from above and below league average K/9 values despite having an above average active spin on their changeup. This variation showed as many data points overlapped and there wasn't really any distinction between clusters. This led to a 0.533 silhouette score which is less than ideal. Ideally we would like to have a 0.7 silhouette score or higher which would indicate our model did well. The silhouette score showed why it is less than ideal in our data visualization as there weren't really any distinguishable clusters.

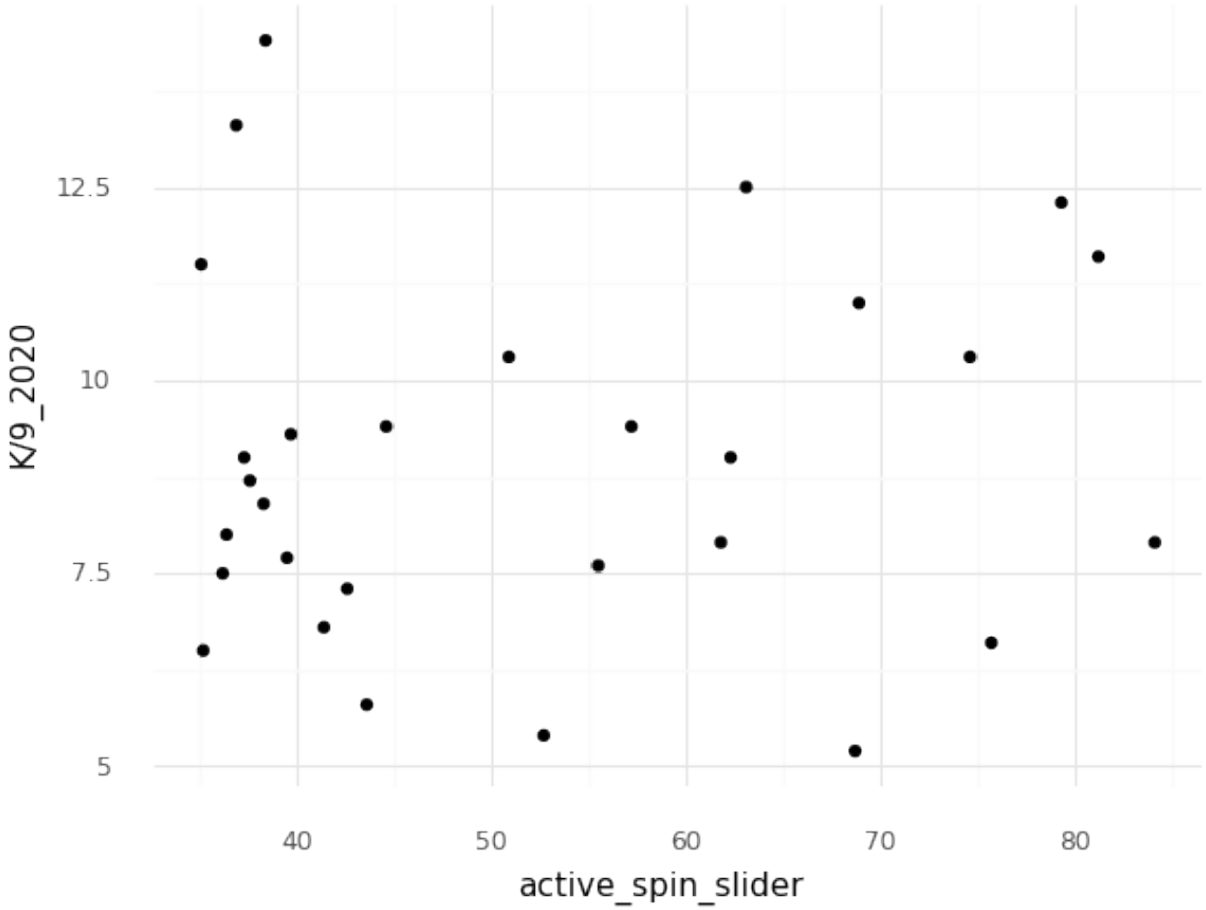
Q5: What groups are formed when looking at pitchers with a slider above 35% active spin and an above average K/9?

In [ ]:
```
HSlider = baseball.loc[baseball.active_spin_slider >= 35]

HSlider.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change
3	Urquidy, Jose	HOU	R	96.7	35.810811	92.800000
7	Bauer, Trevor	CIN	R	95.7	28.300000	87.418182
12	Fiers, Mike	OAK	R	94.3	35.810811	80.700000
17	Thompson, Ryan	TB	R	93.6	35.810811	87.418182
20	Buehler, Walker	LAD	R	93.1	49.500000	87.418182

In [ ]:
```
(ggplot(HSlider, aes("active_spin_slider", "K/9_2020")) + geom_point()
+ theme_minimal())
```



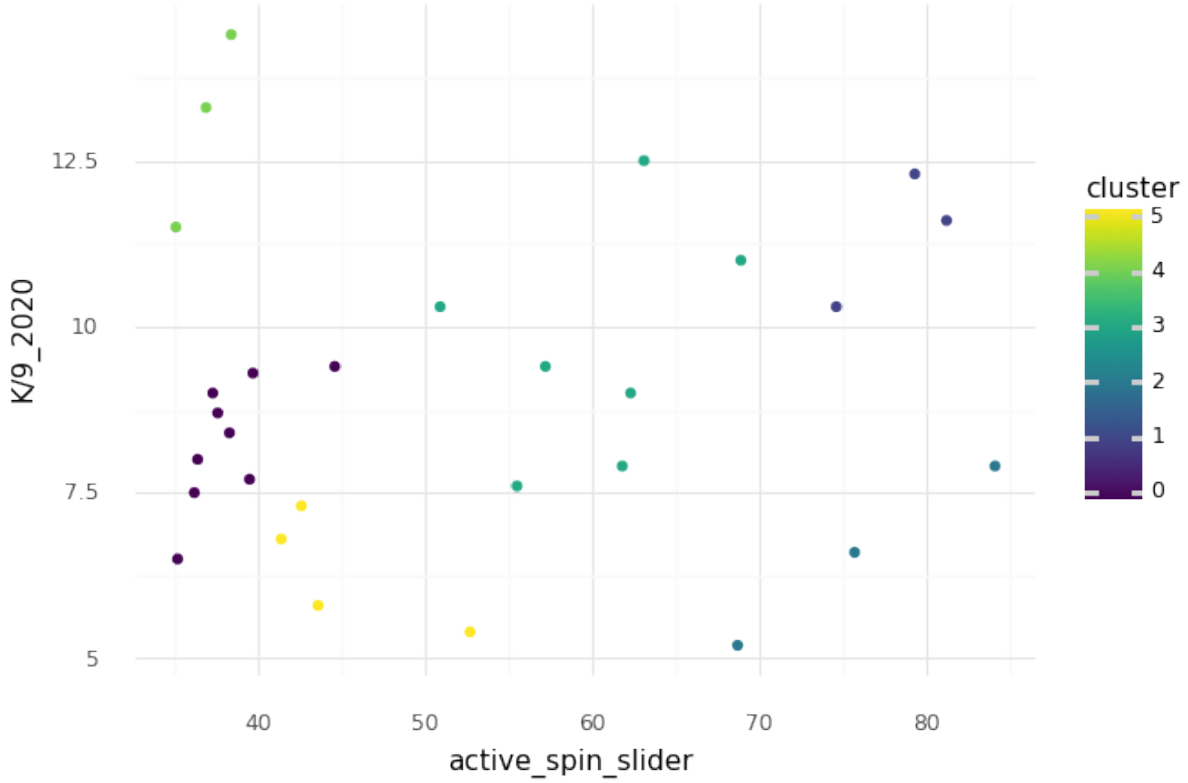
Out[ ]: <ggplot: (-9223363282333017186)>

This graph shows the relationship between the variables K/9 and average active spin of a slider above 35%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["active_spin_slider", "K/9_2020"]

X = HSlider[features]
z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)
EM.fit(X)

cluster = EM.predict(X)

X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]: 0.6567111455323971

In [ ]:
```
(ggplot(HSlider, aes("active_spin_slider", "K/9_2020", color = "cluste
r"))+ geom_point() + theme_minimal())
```



Out[ ]: <ggplot: (8788206486178)>

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph the clusters formed on the left side show pitchers who have an average active spin on a slider of around 35-55 either above league average or below league average K/9 values. The middle cluster which shows pitchers who have an average active spin of 60 on their slider have around league average to above league averageK/9 values. On the right side the clusters formed indicate pitchers who have an average active spin on their slider around 70-80 have below league average or above league average K/9 values.

Answer to question: When looking at pitchers above an average active spin of 35% on their slider and an above average K/9 value groups formed despite variation. All groups had variation, but most groups showed that data points tended to fall around league average K/9 values. The silhouette score we obtained was 0.656 which is decent but not at the 0.7 score we strive to be at. Because our score was slightly below the standard, we can say that our model performed decent but not well enough to trust. In the data visualization we can see the groups but they are not as distinguishable as we want them to be and that could possibly be the reason why we have a less than desirable silhouette score.

Q6: How well does a pitcher's active spin on all of their pitches accurately predict their ERA?

In [5]:
```
predictors = ["active_spin_fastball","active_spin_cutter","active_spin
_change","active_spin_slider","active_spin_curve"]
X_train, X_test, y_train, y_test = train_test_split(baseball[predictor
s], baseball["ERA_2020"], test_size=0.2)
```

In [6]:
```
continuous = ["active_spin_fastball","active_spin_cutter","active_spin
_change","active_spin_slider","active_spin_curve"]
zscore = StandardScaler()
zscore.fit(X_train[continuous])
X_train[continuous] = zscore.transform(X_train[continuous])
X_test[continuous] = zscore.transform(X_test[continuous])
```

In [7]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[7]:
```
LinearRegression()
```

In [8]:
```
y_pred = model.predict(X_test)
y_pred[1:10]
```

Out[8]:
```
array([4.14060082, 4.16154286, 4.12440692, 4.23147164, 4.24012608,
       4.18988263, 4.22775699, 4.07497968, 3.9490547 ])
```

In [9]:
```
model.score(X_test, y_test)
```

Out[9]:
```
-0.054884655589404696
```

In [10]:
```
model.score(X_train, y_train)
```

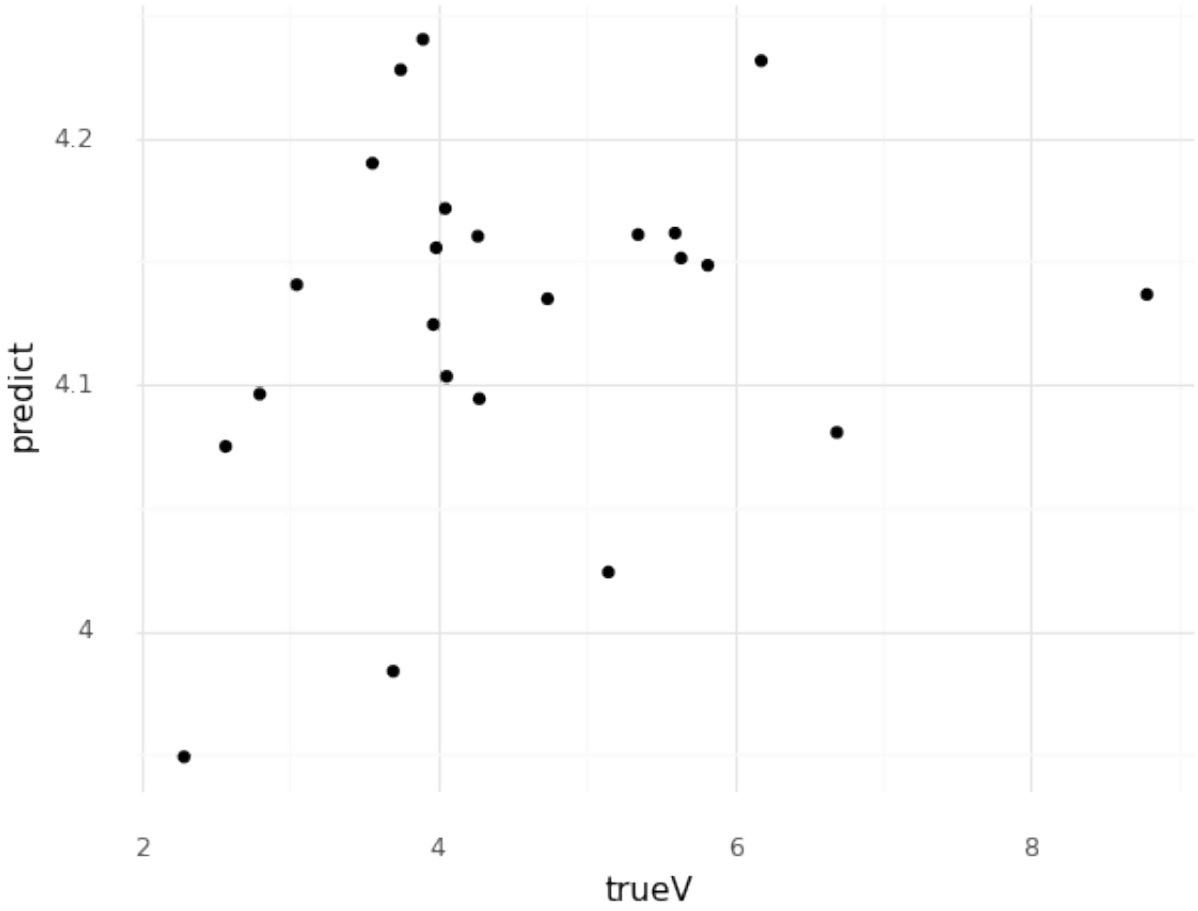
Out[10]:
```
0.004120572892098995
```

In [11]:
```
true_vs_pred = pd.DataFrame({"predict": y_pred,"trueV": y_test})
true_vs_pred.head()
```

Out[11]:

	predict	trueV
84	4.171460	4.05
50	4.140601	3.05
22	4.161543	5.60
28	4.124407	3.97
106	4.231472	6.18

In [12]:
```
(ggplot(true_vs_pred, aes(x = "trueV", y = "predict")) + geom_point()
+ theme_minimal())
```



Out[12]: <ggplot: (306910449)>

This graph shows the relationship between our predicted values and the actual values from the test set. A straight diagonal line would mean that the predicted values and true values are the same, however this graph has a lot of variation and an unclear relationship amongst variables. This means the model might not be very accurate and has a lot of error.

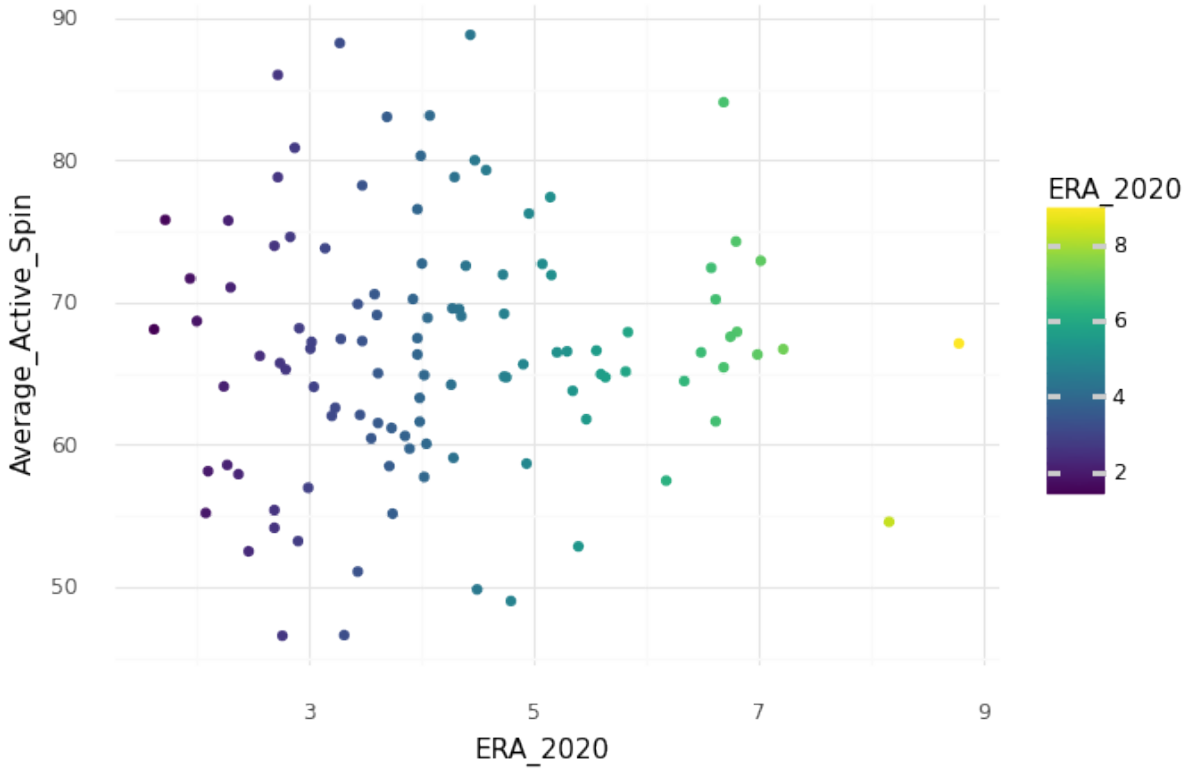
In [13]:
```
coefficients = pd.DataFrame({"Coef":model.coef_,
"Name": predictors})
coefficients = coefficients.append({"Coef": model.intercept_,
"Name": "intercept"}, ignore_index = True)
```

In [14]:
```
coefficients
```

Out[14]:

	Coef	Name
0	-0.023293	active_spin_fastball
1	0.055084	active_spin_cutter
2	-0.043688	active_spin_change
3	-0.032218	active_spin_slider
4	-0.036478	active_spin_curve
5	4.124607	intercept

In [15]:
```
(ggplot(baseball, aes(x = "ERA_2020", y = "Average_Active_Spin", fill
= "ERA_2020", color = "ERA_2020"))+ geom_point() +
theme_minimal())
```



Out[15]: <ggplot: (306914033)>

This graph shows the relationship between a pitcher's active spin on a slider and their ERA. We wanted to specify a variable to graph to show the relationship to ERA after running it through the linear regression model. We can see that pitchers with a high active spin on their slider has pretty average ERA values and pitcher with low active spin on their slider also had pretty average ERA, however a solid group of pitchers with low active spin also showed a higher ERA. We can conclude that a pitcher's active spin on all their pitches is not a great predictor of ERA.

Answer to Question: After running the linear regression model, we can conclude that a pitcher's active spin on all of their pitches did not very accurately predict their BB/9. After outputting the r2 value for the train and test set, we can see that they were very low values and we want the r2 values to be high. This means the model had low accuracy. After plotting the true v predicted values we can see there was high variation and not a clear relationship between the data. This is an indicator of a good amount of error in the model. We can conclude that the active spin of a pitcher's arsenal isn't the best predictor of BB/9 and should be reevaluated to find better accuracies.

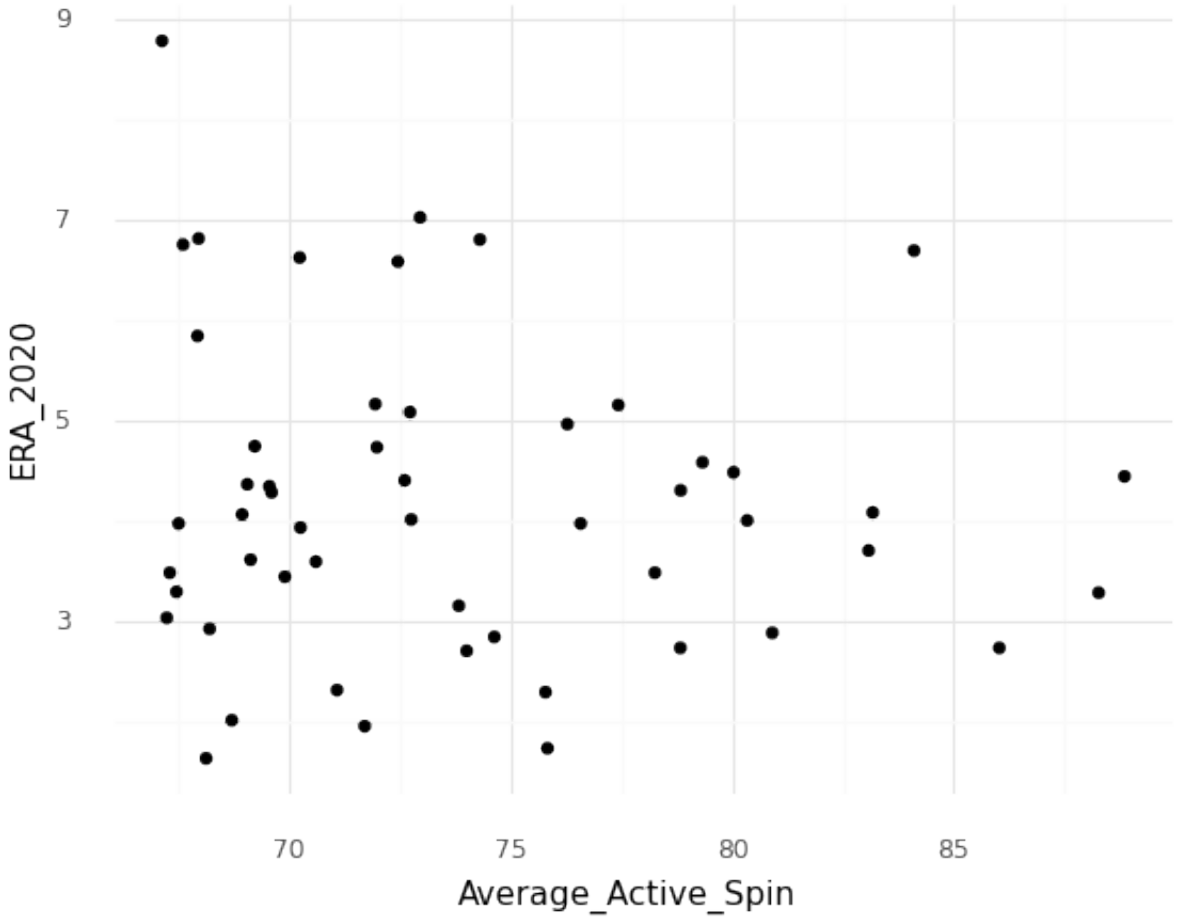
Q7: What groups are formed when looking at pitchers with an average active spin above 67% and their ERA?

In [ ]:
```
ActSpin = baseball.loc[baseball.Average_Active_Spin >= 67]
ActSpin.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change	ac
0	Mahle, Tyler	CIN	R	99.9	35.810811	88.1	
1	Cole, Gerrit	NYY	R	99.9	35.810811	99.0	
3	Urquidy, Jose	HOU	R	96.7	35.810811	92.8	
4	Gonsolin, Tony	LAD	R	96.6	35.810811	90.1	
5	Cobb, Alex	BAL	R	96.5	35.810811	73.9	

In [ ]:
```
(ggplot(ActSpin, aes("Average_Active_Spin", "ERA_2020")) + geom_point(
) + theme_minimal())
```



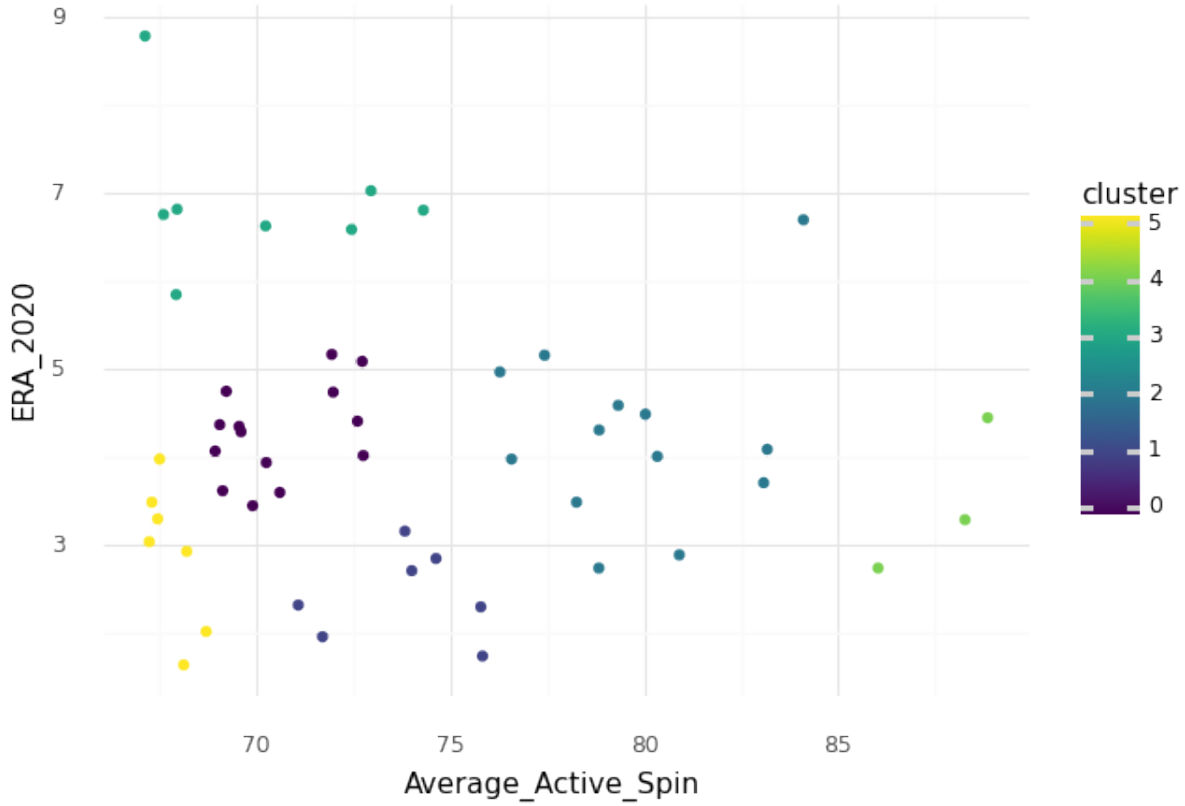
Out[ ]: <ggplot: (8754521629388)>

This graph shows the relationship between the variables ERA and average active spin above 67%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["Average_Active_Spin", "ERA_2020"]

X = ActSpin[features]
z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)
EM.fit(X)

cluster = EM.predict(X)

X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]:
```
0.6386321050001615
```

In [ ]:
```
(ggplot(ActSpin, aes("Average_Active_Spin", "ERA_2020", color = "clust
er"))+ geom_point() + theme_minimal())
```



Out[ ]:
```
<ggplot: (-9223363282333075133)>
```

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph the clusters formed on the left side show pitchers who have an average active spin of around 70 to have ERA numbers ranging from below league average to above league average. The middle clusters that indicate the pitcher has an average active spin of 75-80 show that they tend to have around league average to below league average ERA numbers. The left side cluster that shows pitchers who have an average active spin of around 85 tend to also have around league average to below league average ERA numbers. This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph the clusters formed on the left side show pitchers who have an average active spin of around 50 have around league average to below league average ERA numbers. The middle cluster indicates pitchers who have an average active spin of around 55 have below league average ERA numbers. The right side clusters show pitchers who have an average active spin of around 60-67 have ERA numbers ranging from below league average to above league average.

Answer to Question: When looking at pitchers with an average active spin above 67% and their ERA, groups did form despite a good amount of variation. Although there was variation, the groups tended to form around league average ERA values despite the differences in average active spin. We can see that pitchers with low active spin still had league average ERA values and pitchers with high active spin had general league average ERA values as well. With a silhouette score of about 64%, we can say that our model did somewhat well in the clustering of our data, however we would've liked it to be higher. In the data visualization we can see the groups but they are not as distinguishable as we want them to be and that could possibly be the reason why we have a less than desirable silhouette score.

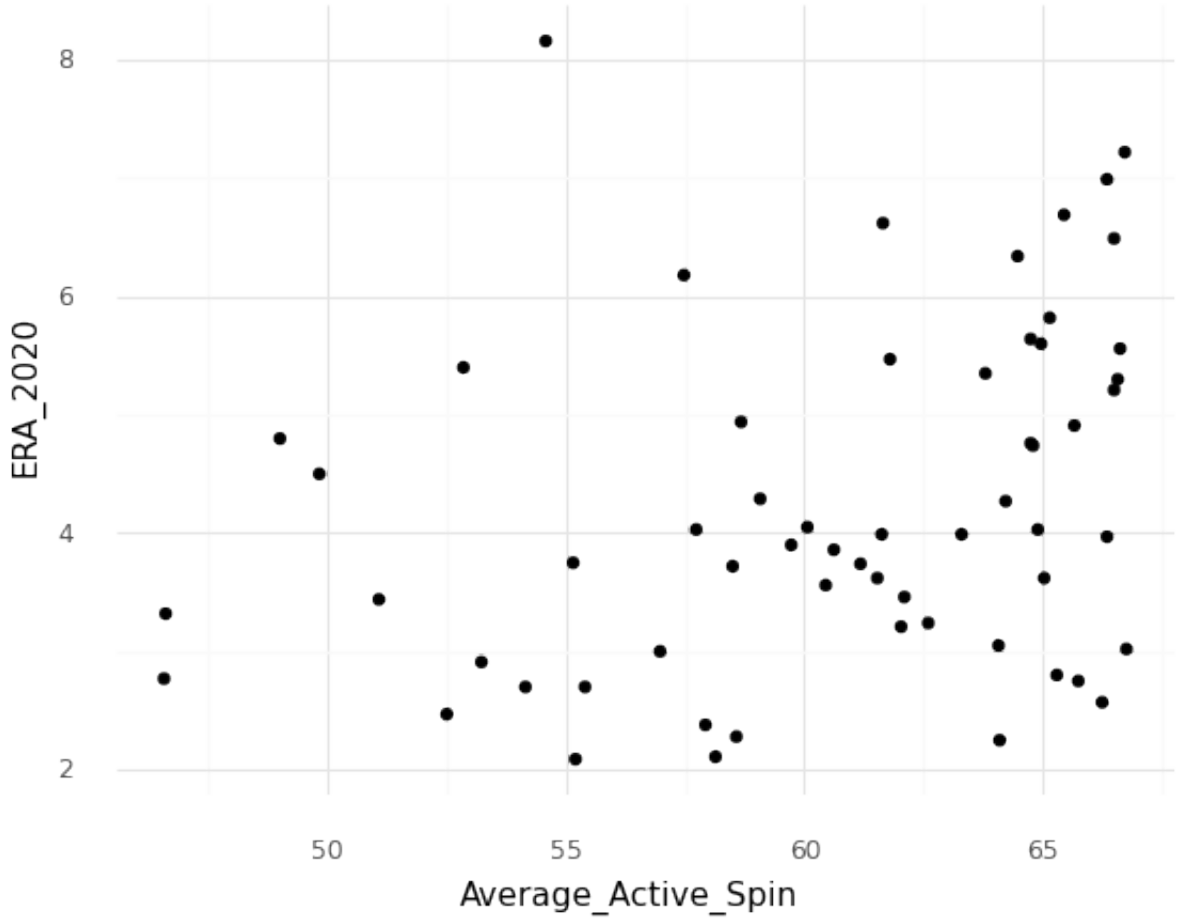
Q8: What groups are formed when looking at pitchers with an average active spin below 67% and their ERA?

In [ ]:
```
BActSpin = baseball.loc[baseball.Average_Active_Spin <= 67]
BActSpin.head()
```

Out[ ]:

	Pitcher	Team	throw_hand	active_spin_fastball	active_spin_cutter	active_spin_change
2	May, Dustin	LAD	R	97.7	30.500000	83.7
8	McKenzie, Triston	CLE	R	95.2	35.810811	78.5
14	Quantrill, Cal	CLE	R	93.9	35.810811	77.0
16	Canning, Griffin	LAA	R	93.8	35.810811	94.6
22	Montas, Frankie	OAK	R	92.9	35.810811	72.6

In [ ]:
```
(ggplot(BActSpin, aes("Average_Active_Spin", "ERA_2020")) + geom_point() + theme_minimal())
```



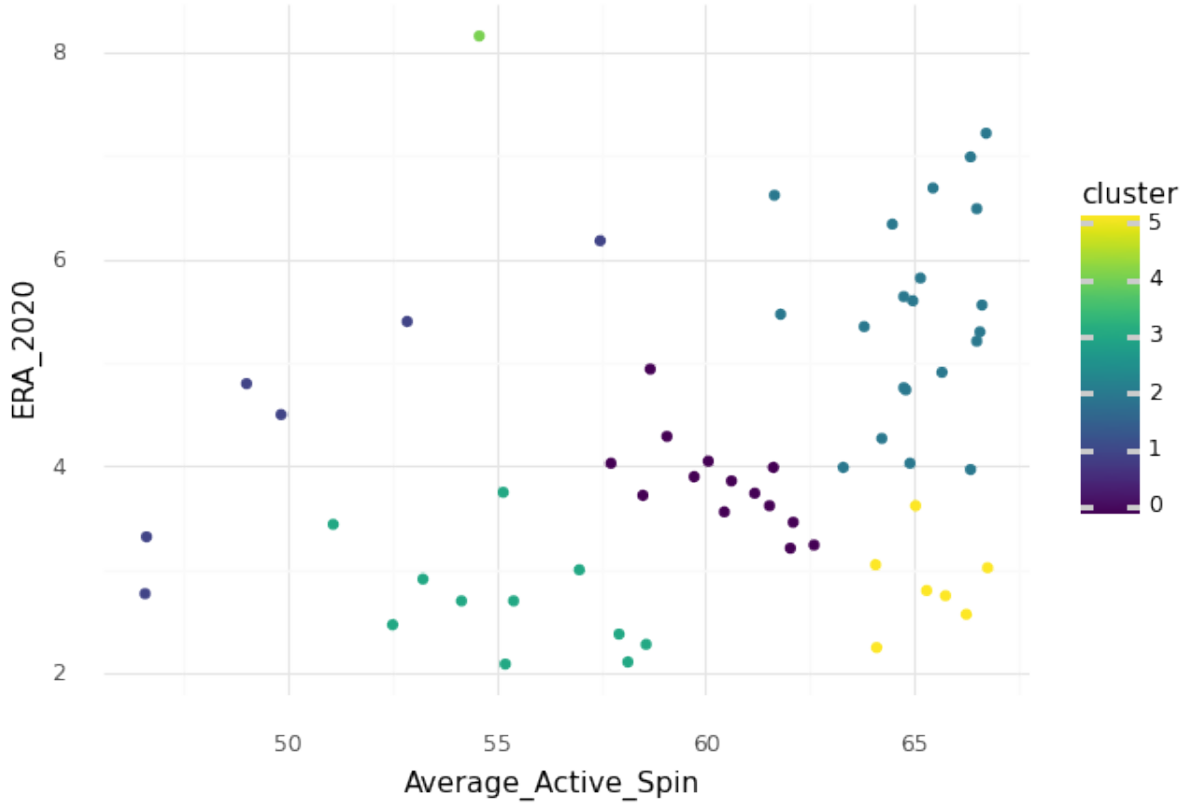
Out[ ]: <ggplot: (8754521699960)>

This graph shows the relationship between the variables ERA and average active spin below 67%. The purpose of the graph is to show the general layout of the data and the relationships of the variables before we anaylze more in depth with the expecation maximization clustering model.

In [ ]:
```
features = ["Average_Active_Spin", "ERA_2020"]

X = BActSpin[features]
z = StandardScaler()

X[features] = z.fit_transform(X)

EM = GaussianMixture(n_components = 6)
EM.fit(X)

cluster = EM.predict(X)

X["cluster"] = cluster
silhouette_score(X,cluster)
```

Out[ ]:
```
0.6658203775841408
```

In [ ]:
```
(ggplot(BActSpin, aes("Average_Active_Spin", "ERA_2020", color = "clus
ter"))+ geom_point() + theme_minimal())
```



Out[ ]: <ggplot: (-9223363282333190638)>

This graph shows the formed clusters of the data after being run through the expectation maximization clustering algorithm. From the graph the clusters formed on the left side show pitchers who have an average active spin of around 50 have around league average to below league average ERA numbers. The middle cluster indicates pitchers who have an average active spin of around 55 have below league average ERA numbers. The right side clusters show pitchers who have an average active spin of around 60-67 have ERA numbers ranging from below league average to above league average.

Answer to Question: When looking at pitchers with an average active spin below 67% and their ERA, groups did form despite a good amount of variation. Although there was variation, the groups tended to form around league average ERA values despite the differences in average active spin. Pitchers with active spin around 60-65 had fairly average league ERA values, however pitchers with active spin around 45-55 had lower than average league ERA values. With a silhouette score of about 66%, we can say that our model did decently well, however we would've liked it to be higher. We can conclude that a pretty clear group that formed was that pitchers with lower active spin values also had lower ERA values.

Q9: How many variables are necessary in predicting a pitcher's BB/9?

In [4]:
```
features = baseball.columns[4:12]
baseball.head()

z = StandardScaler()
baseball[features] = z.fit_transform(baseball[features])
```

In [5]:
```
pca = PCA()

pca.fit(baseball[features])
```

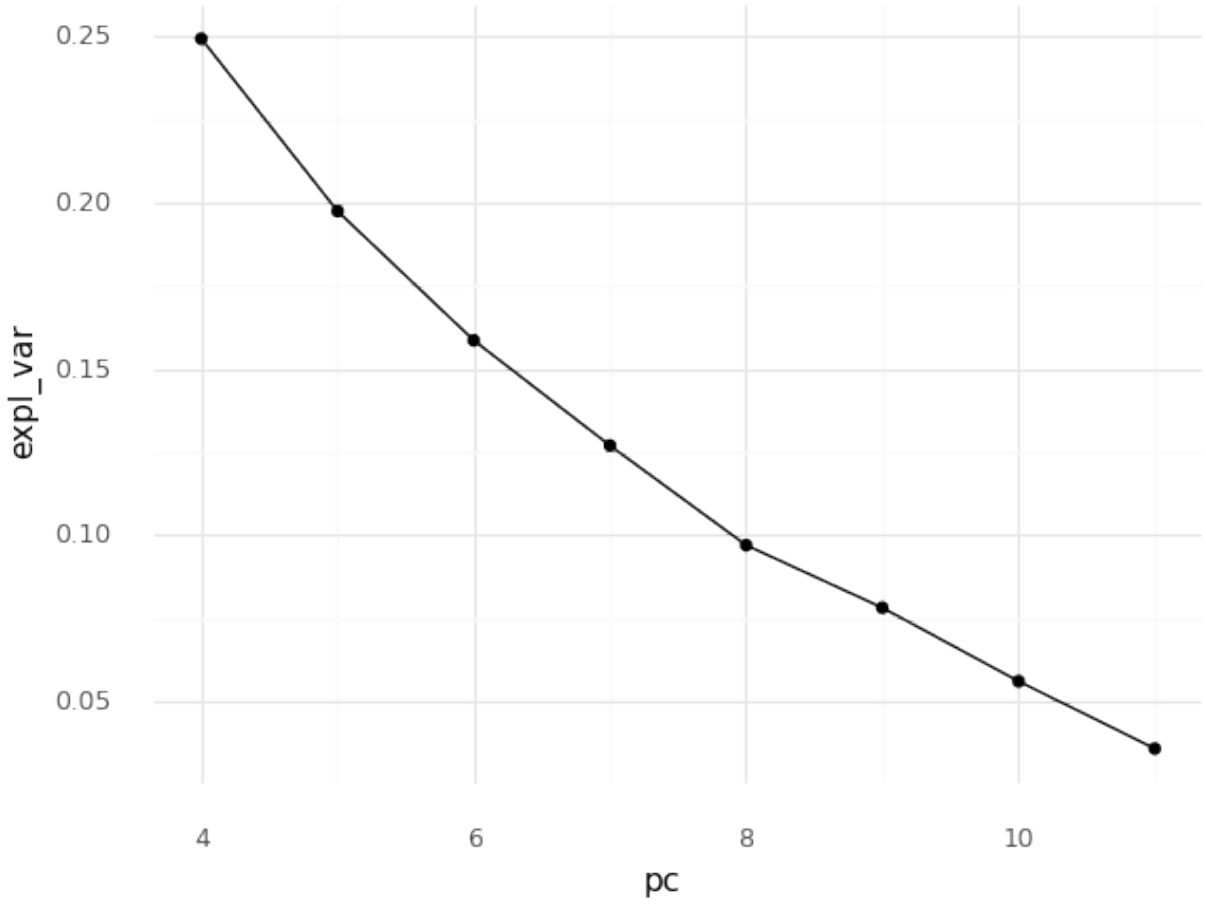
Out[5]:
```
PCA()
```

In [6]:
```
pcaDF = pd.DataFrame({"expl_var" : pca.explained_variance_ratio_,
                      "pc": range(4,12), "cum_var": pca.explained_vari
ance_ratio_.cumsum()})
pcaDF.head()
```

Out[6]:

	expl_var	pc	cum_var
0	0.249320	4	0.249320
1	0.197508	5	0.446827
2	0.158669	6	0.605496
3	0.127041	7	0.732537
4	0.097080	8	0.829617

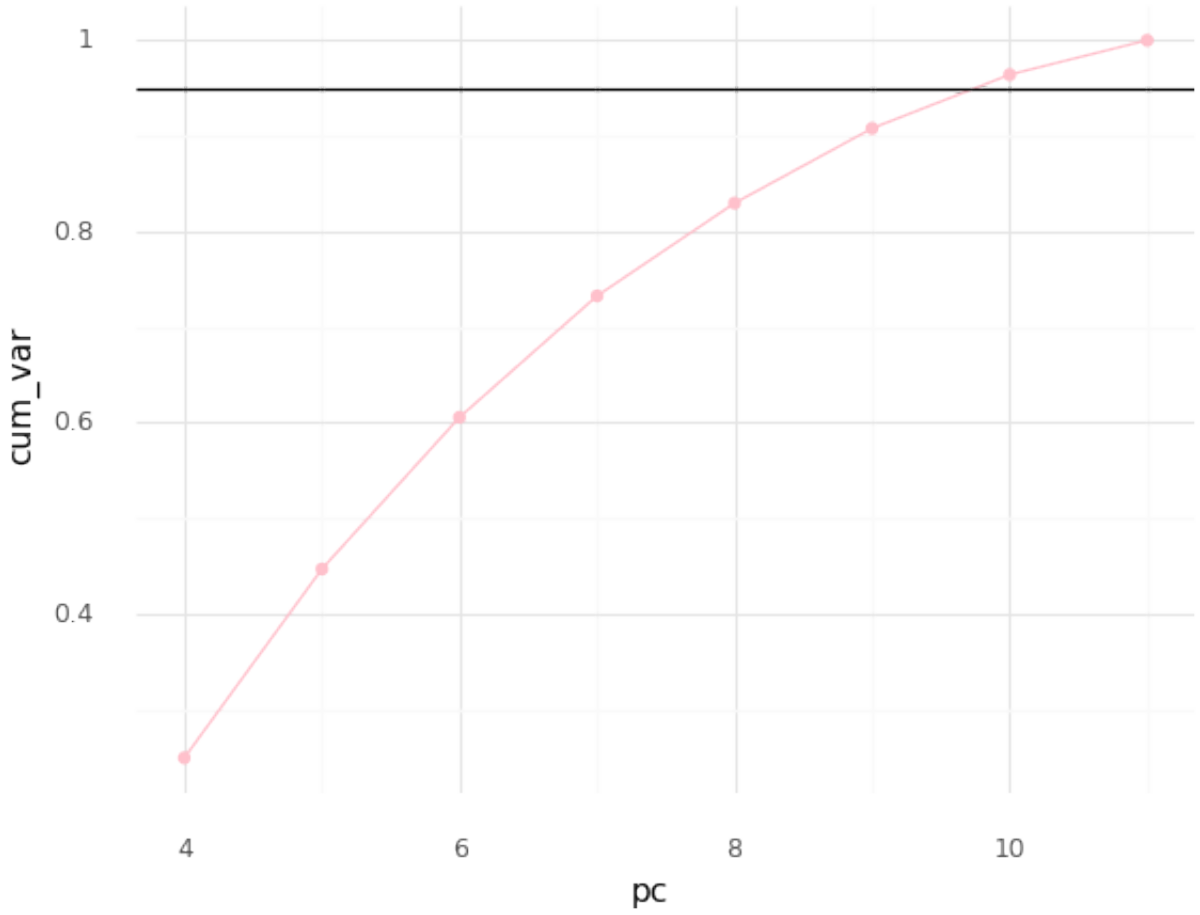
In [7]:
```
ggplot(pcaDF, aes(x = "pc", y = "expl_var")) + geom_line() + geom_poin
t() + theme_minimal()
```



Out[7]: <ggplot: (283353941)>

This ggplot graph is a screeplot graph that has explained variance on the y-axis and the principle components on the x-axis. We can see that the first component accounts for about 25% of the variance, the second component accounts for about 20% of the variance, the third component accounts for about 16% of the variance, the fourth component accounts for about 13% of the variance, and the fifth component accounts for about 10% of the variance. After the 5th component we can see that the rest of the components are not accounting for a lot of variance and should be considered for removal from the model.

In [8]:
```
(ggplot(pcaDF, aes(x = "pc", y = "cum_var")) + geom_line(color = "pink
")+
 geom_point(color = "pink")+ geom_hline(yintercept = 0.95) + theme_min
imal())
```



Out[8]: <ggplot: (307452697)>

This graph is another form of a scree plot. This graph is looking at, across all principle components, how much of the total variation is explained by the PCs cumulatively. The first two PCs describe about 45% of the variation, the first three PCs describe about 60% of the variation, the first four PCs describe about 75% of the variation, and the first five PCs describe about 83% of the variation. We chose to have our PCs describe at least 95% of the information in the data. The black line at 95% represents this, and we can see that the first six components explain a little less than 95% of the variation that we had out of all the original variables. We can conclude that we should keep six variables out of all of them because they explain the majority of the variation.

In [9]:
```
pcomps4 = pca.transform(baseball[features])
pcomps4 = pd.DataFrame(pcomps4[:,1:4])


pcomps6 = pca.transform(baseball[features])
pcomps6 = pd.DataFrame(pcomps6[:, 1:6])

#modeMod1
lr1 = LinearRegression()
lr1.fit(baseball[features], baseball["BB/9_2020"])
print("all data: ", lr1.score(baseball[features], baseball["BB/9_2020"
]))

#modeMod1
lr2 = LinearRegression()
lr2.fit(pcomps6, baseball["BB/9_2020"])
print("6 PCs:   ", lr2.score(pcomps6, baseball["BB/9_2020"]))

#modeMod1
lr3 = LinearRegression()
lr3.fit(pcomps4, baseball["BB/9_2020"])
print("4 PCs:    ", lr3.score(pcomps4, baseball["BB/9_2020"]))
```

```
all data:  1.0
6 PCs:    0.8921231911851697
4 PCs:     0.8636556648397311
```

In this code, we pulled some of the PCs from the data. We pulled the first four PCs and the first 6 PCs. We then built a linear regression models that predicts a pitcher's BB/9 based on the active spin on all of their pitches using either all of the data, the first 6 PCs, or the first 4 PCs. The code then outputs the accuracy of each model. When using 6 PCs, we can see that the accuracy of about 89% is high, and using 4PCs the accuracy at 86% is also high. We can conclude that by sacrificing some variables and only using a few PCs, the accuracy is still high and we are still retaining a lot of information from the data.

Answer to Question: After running the principal component analysis, we can conclude that 6 variables/PCs are necessary in predicting a pitcher's BB/9. We wanted our PCs to describe at least 95% of the information in the data. Out of all of our variables, using 6 PCs brought us to just under 95% explained variation. The scree plots visualized this, where the cumulative PCs crossed the 95 % black line indicator at just about 6 PCs.