What do buyers value in beef calf and feeder cattle lots?

Esther McCabe

GENBA 894

August 5, 2019

About the Dataset

- Stored and maintained in Access
- Started in 1995
  - Maintained by one individual
- 241,278 lots total (26,435,097 head of cattle)
  - Cows, bulls, calves, feeders, dairy cattle

Variables

- **SYEAR**: Sale Year 1995 - 2018
- **LOTID**: Count of number of lots of beef calves
- **SMONTH**: Sale Month 1=January, 12=December
- **SUMMER**: Sold in a summer sale, all 1=Yes
- **HEAD**: Number of calves in a lot
- **SEX**: Gender of lot, 1=Steer, 2=Heifer
- **ATYPE**: Animal Type, 0=Unweaned, 1=Weaned
- **WT**: Average weight of the lot (total lot weight/# of head in lot)
- **PRICE**: Sale Price of lot ($/cwt)
- **STATE**: State of origin of lot
- **STATECODE**: Code of state, alphabetical order starting with 1
- **SAREA**: Sale area, the states are divided into five areas or regions, 1=West Coast, 2=Rocky Mountain/North Central, 3=South Central, 4=Northeast, 5=Southeast
- **BREED**: General breed of the lot, 1=English, English crossed, 3=English-Conteninal crossed, 4=Brahman-influenced
- **FRAME**: Frame score of calves in lot, 3=Small, 4=Medium, 5=Large
- **FLESH**: Amount of flesh (body condition) of calves in lot, 2=Light, 3=Light/Medium, 4=Medium, 5=Heavy
- **VAC**: Vaccinations of lot, 0=vaccinated but not qualified for program, 1=vaccinated and qualify for program

Data Transformation and Cleaning

Data Transformation and Cleaning

Lots of beef calves

- No missing variables
- 94,872 lots
- Removed unnecessary variables
  - Summer
  - ATYPE
- Grouped similar, small groups with larger groups within variables

Lots of feeder cattle

- No missing variables
- 23,862 lots
- Removed unnecessary variables
  - Summer
  - ATYPE
- Grouped similar, small groups with larger groups within variables

Exploratory Data Analysis and Visualization

Descriptive Statistics

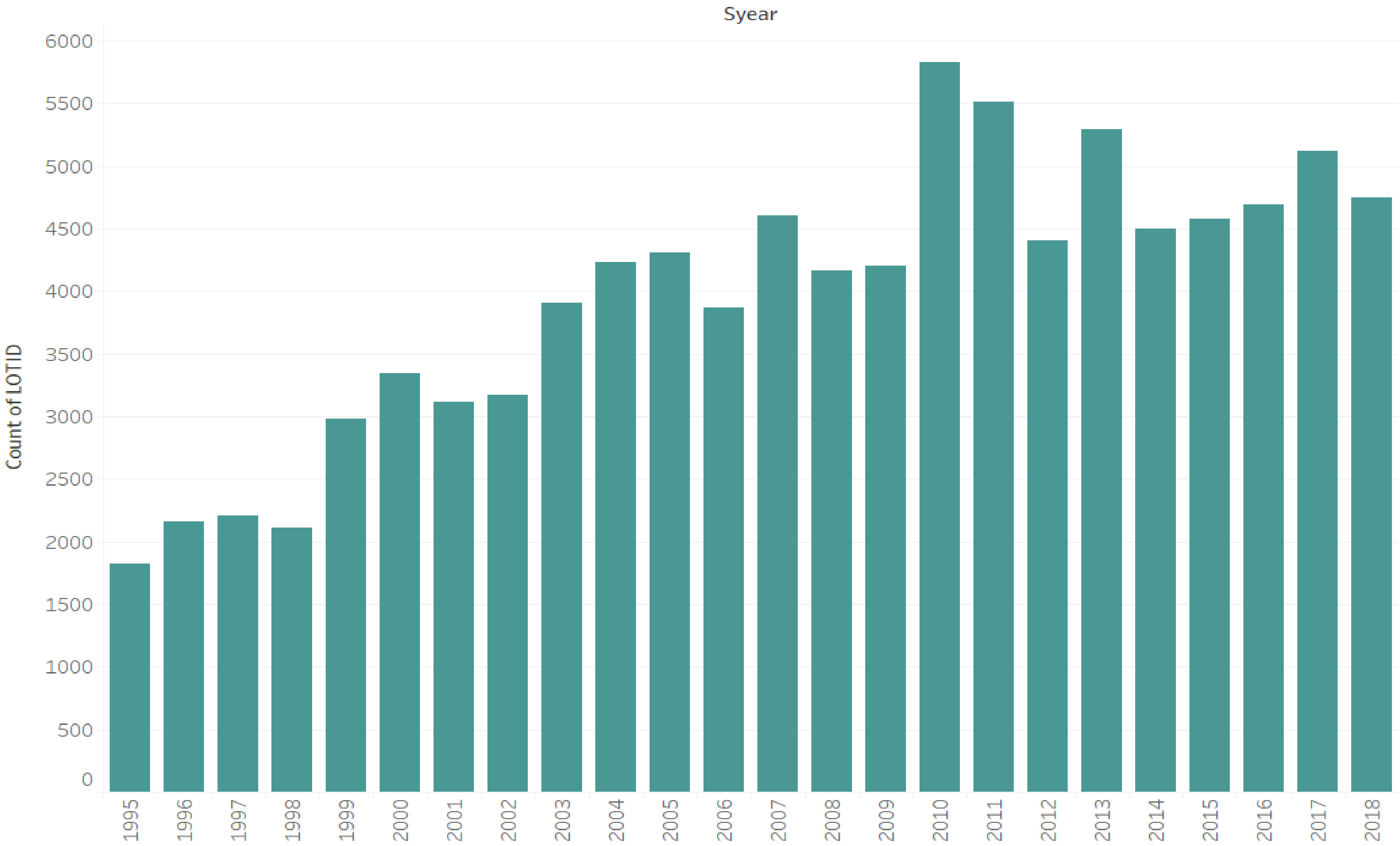
Beef calf lots

Factor	Mean	SD	Range
Size of lot (head)	111.5	72.0	5 to 1,380
Weight	561.0	79.0	235 to 960
Sale Price ($/cwt)	134.75	48.86	41.25 to 422.00

Feeder cattle lots

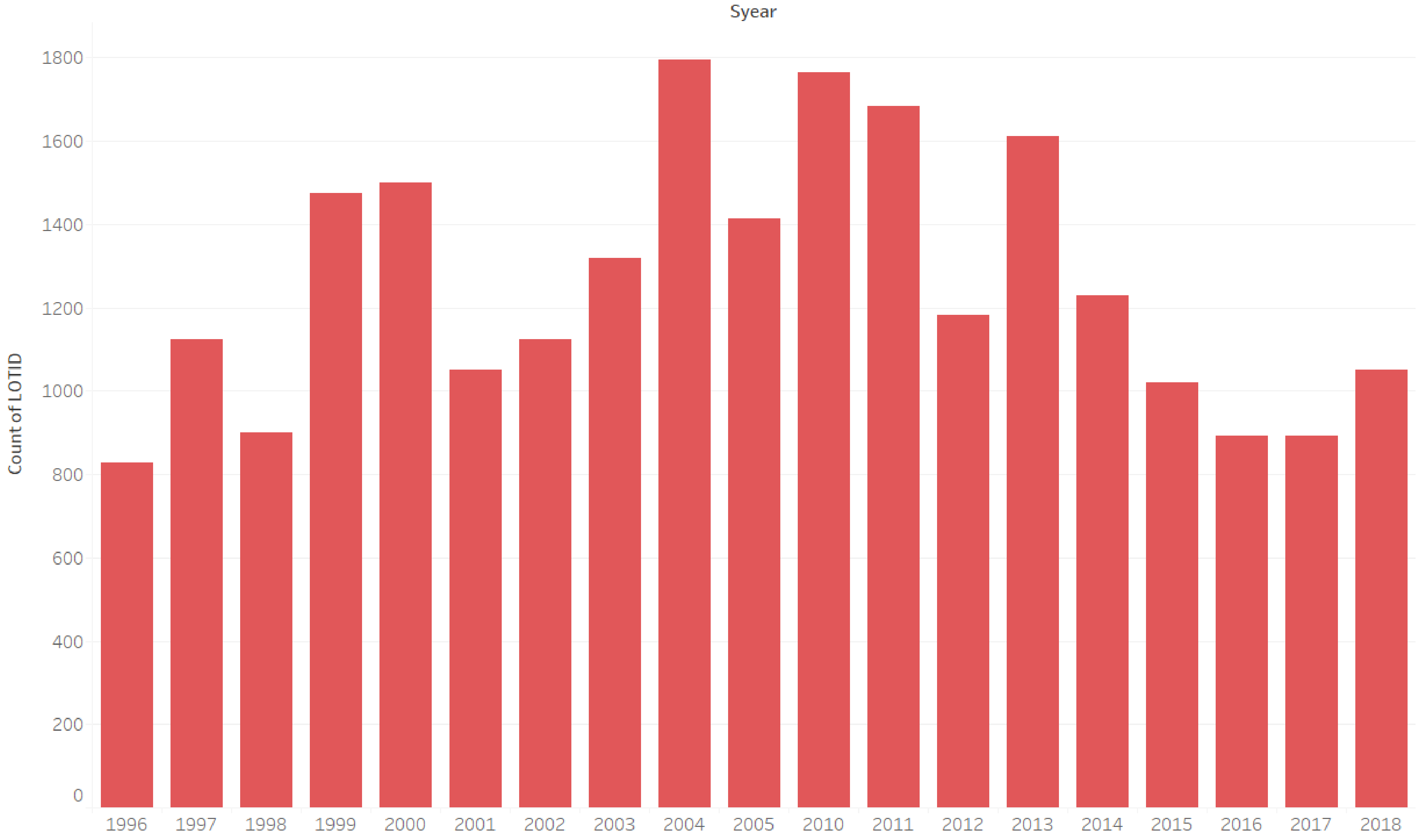
Factor	Mean	SD	Range
Size of lot (head)	125.9	115.5	4 to 1,800
Weight	783.1	84.9	420 to 1,125
Sale Price ($/cwt)	114.91	40.28	45.00 to 267.00

Sale Year

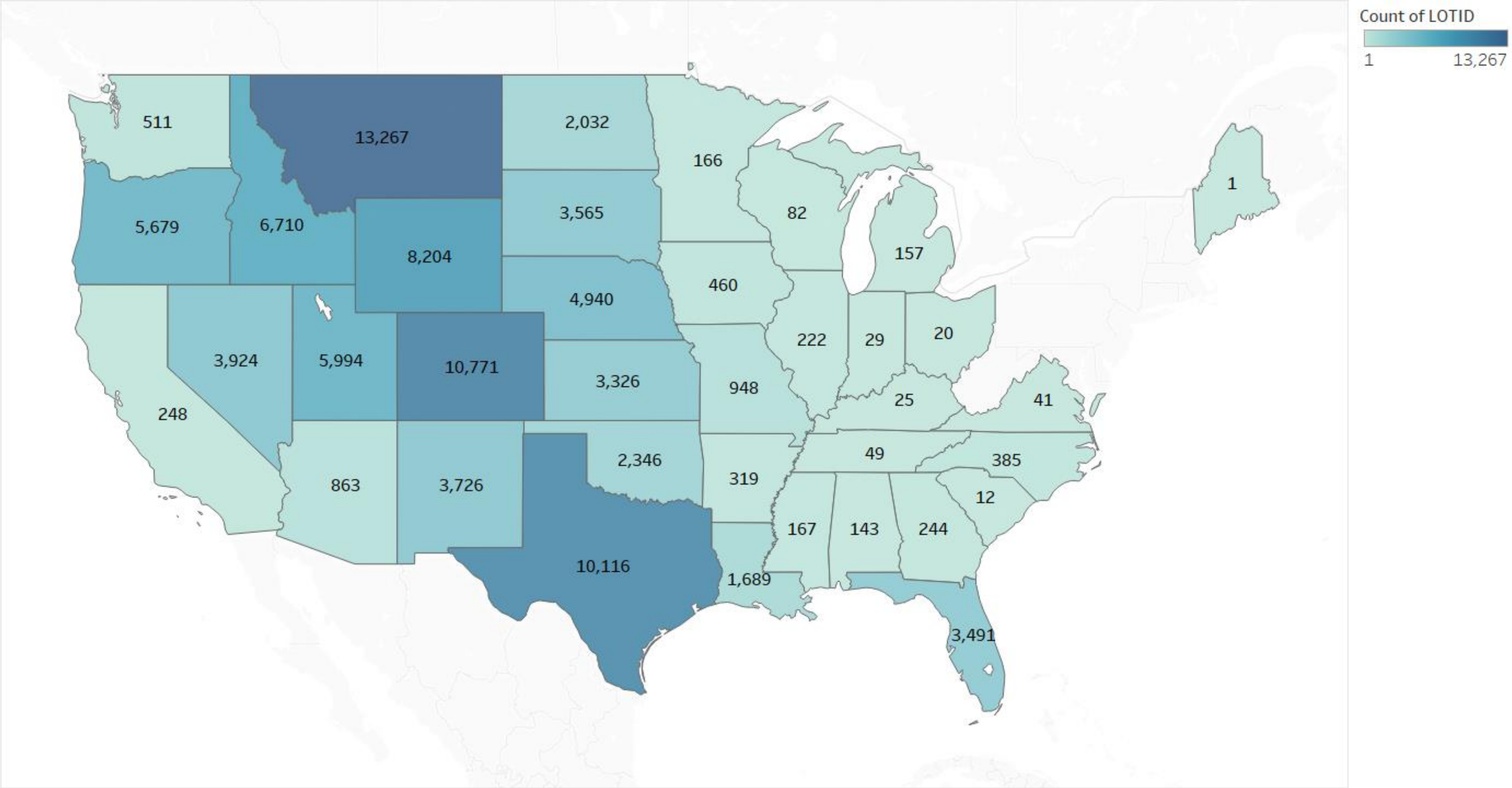


Syear
Count of LOTID
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018

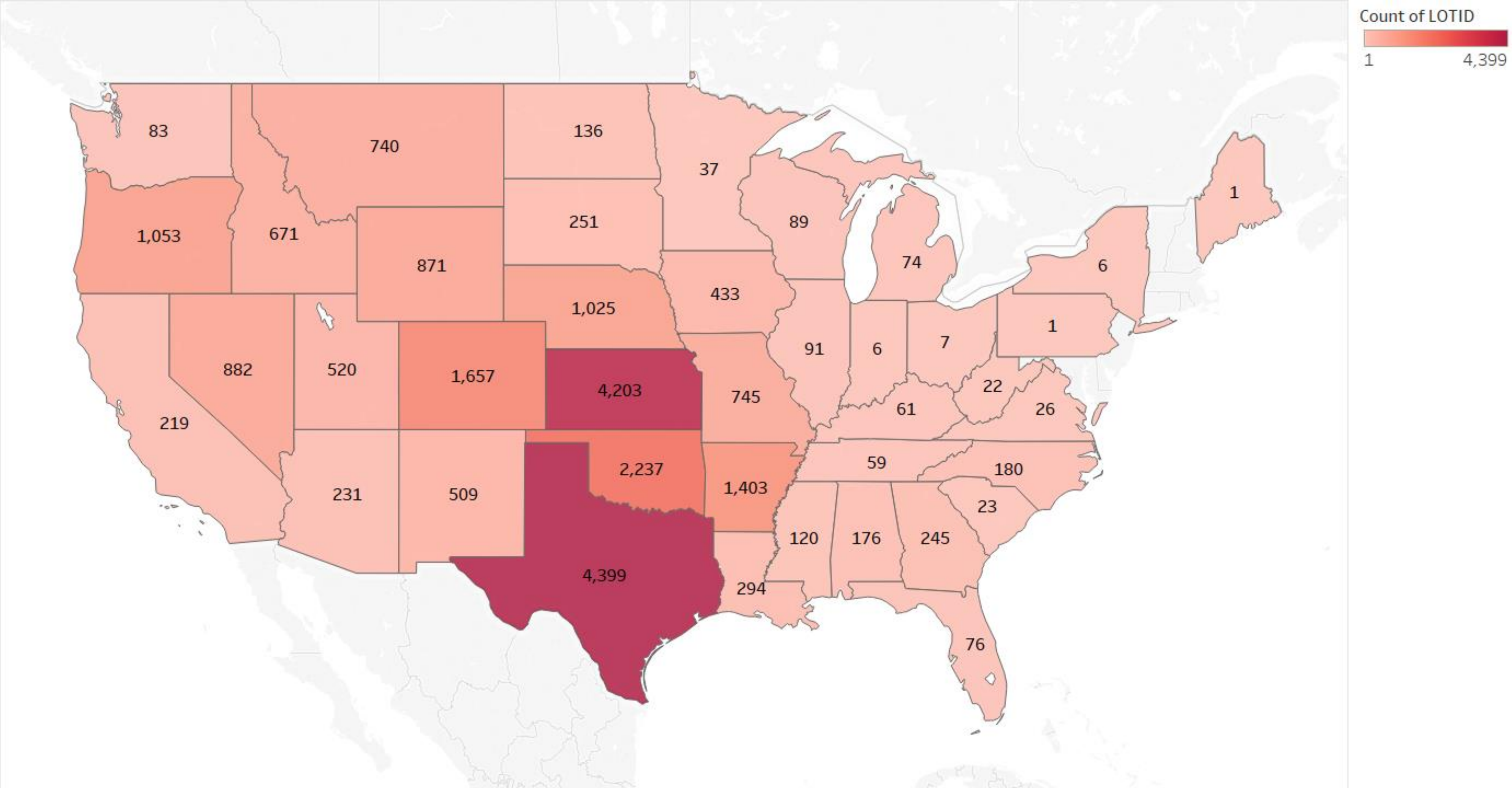
Sale Year


Syear
Count of LOTID
1800
1600
1400
1200
1000
800
600
400
200
0
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2010
2011
2012
2013
2014
2015
2016
2017
2018

State


Count of LOTID
1
13,267
511
13,267
2,032
166
1
5,679
6,710
3,565
82
8,204
157
460
4,940
222
29
20
3,924
5,994
10,771
3,326
948
25
41
248
49
2,346
385
863
3,726
319
12
167
143
244
10,116
1,689
3,491

State

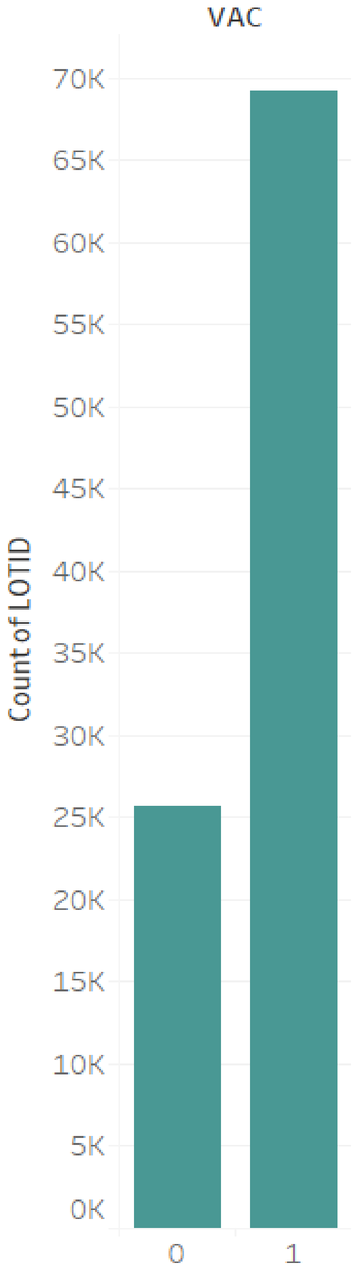


Count of LOTID
1
4,399
83
740
136
37
1,053
671
251
89
1
871
74
6
433
1,025
1
91
6
7
882
520
1,657
4,203
745
22
61
26
219
2,237
59
180
1,403
231
509
23
120
176
245
4,399
294
76

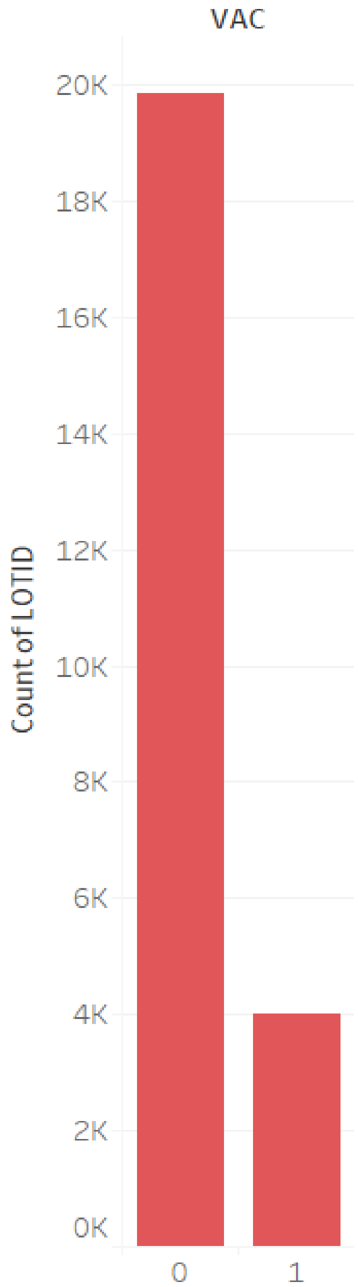
Comparison of Beef Calves and Feeder Cattle Lots

Vaccination Program Qualification

Calves

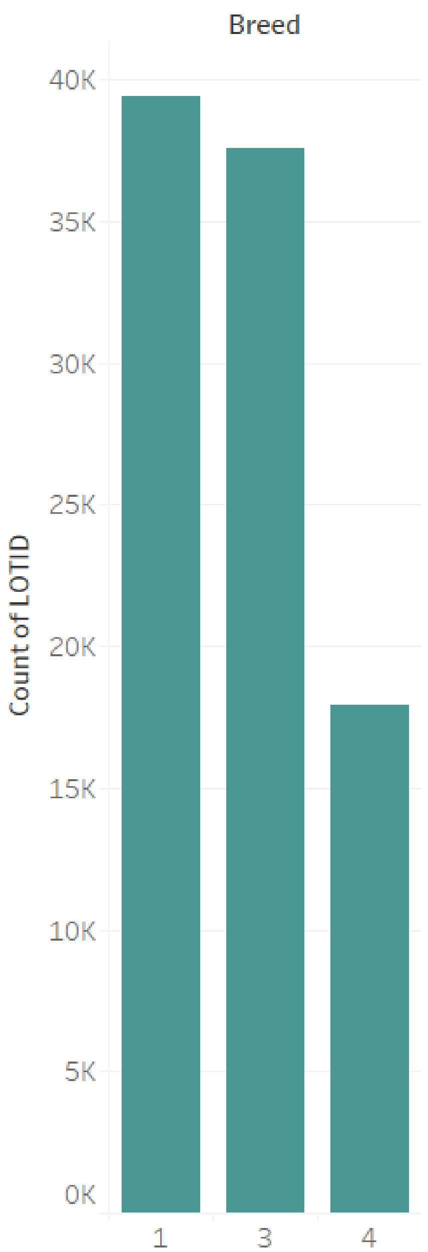


Feeders

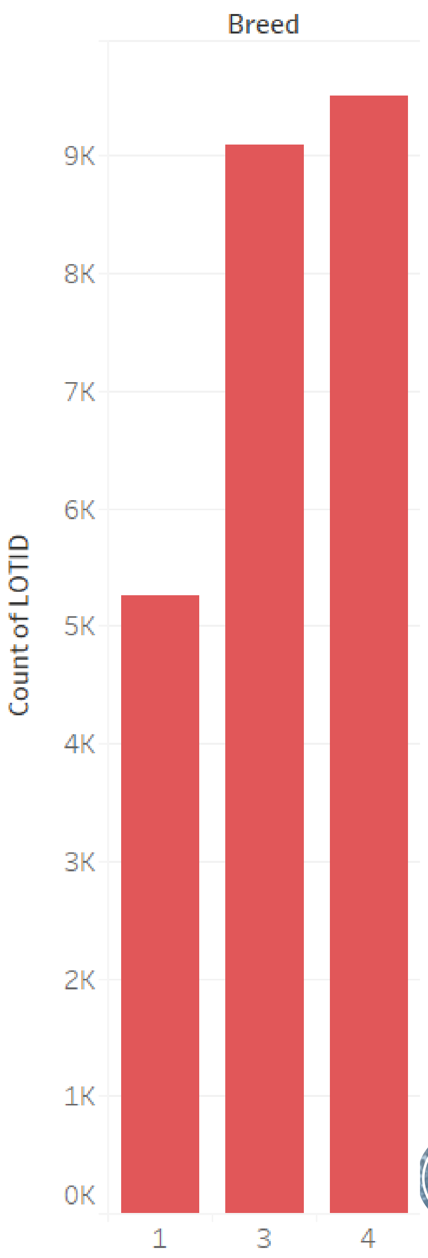


Breed Description

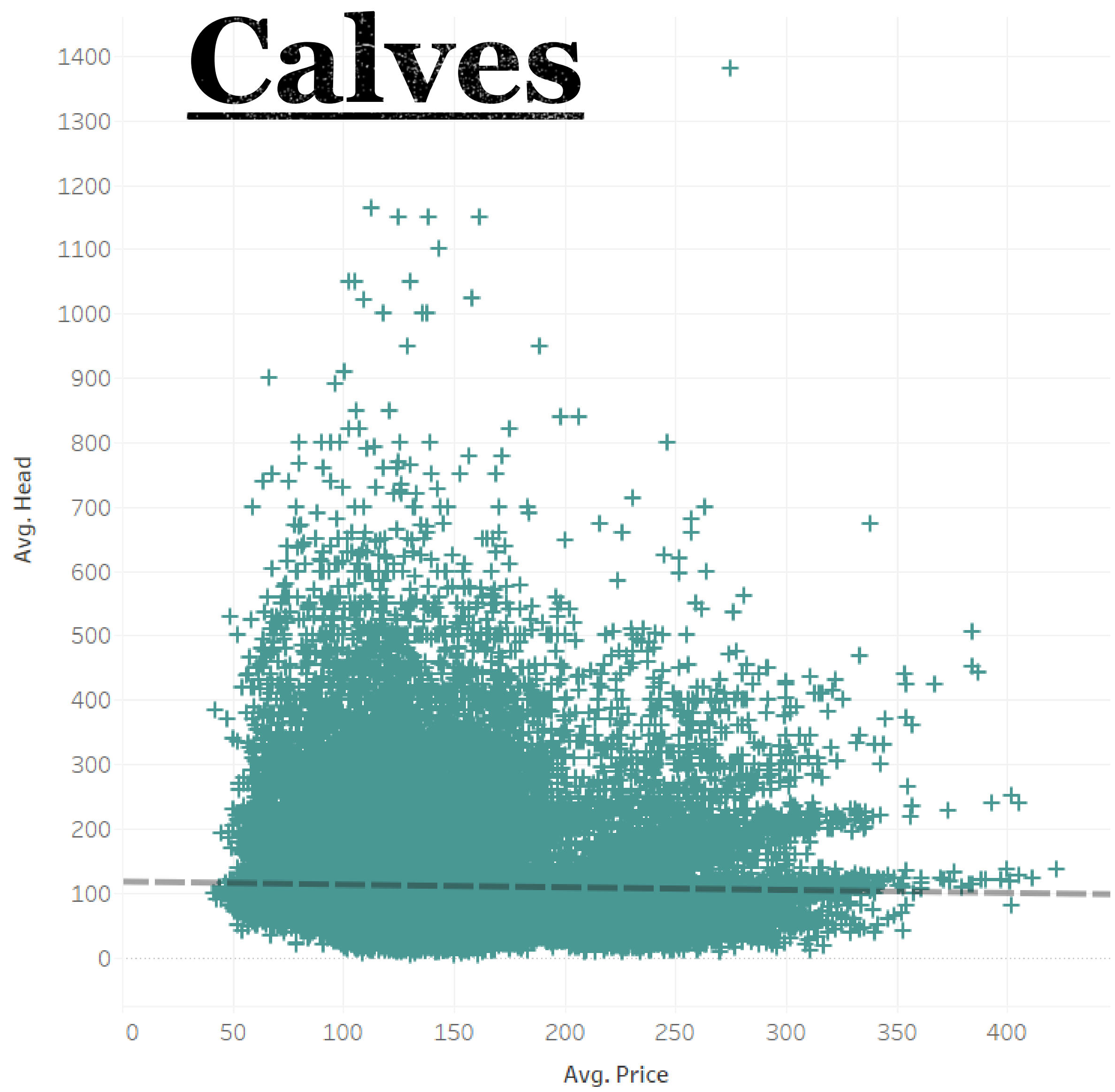
Calves

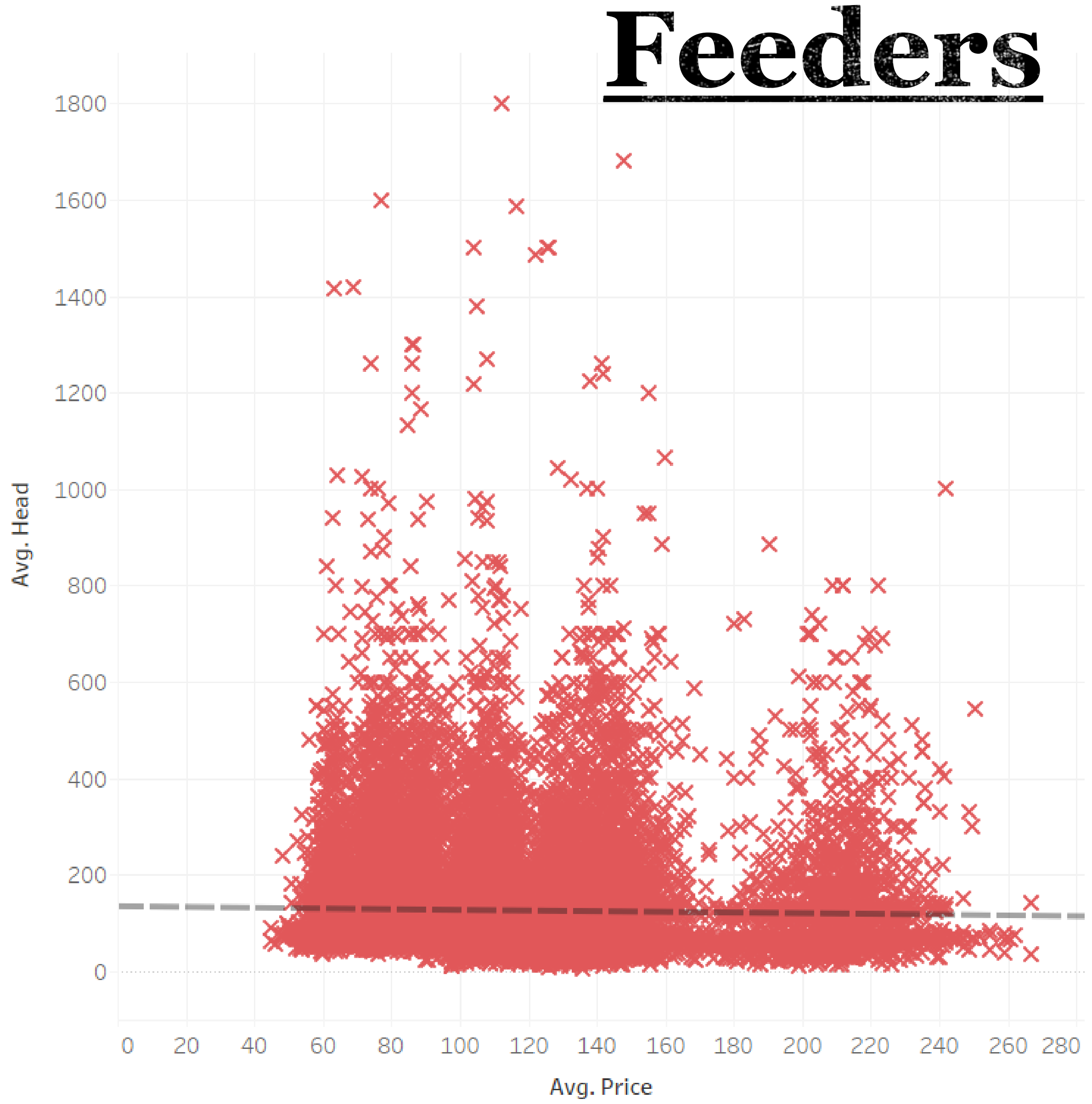


Feeders



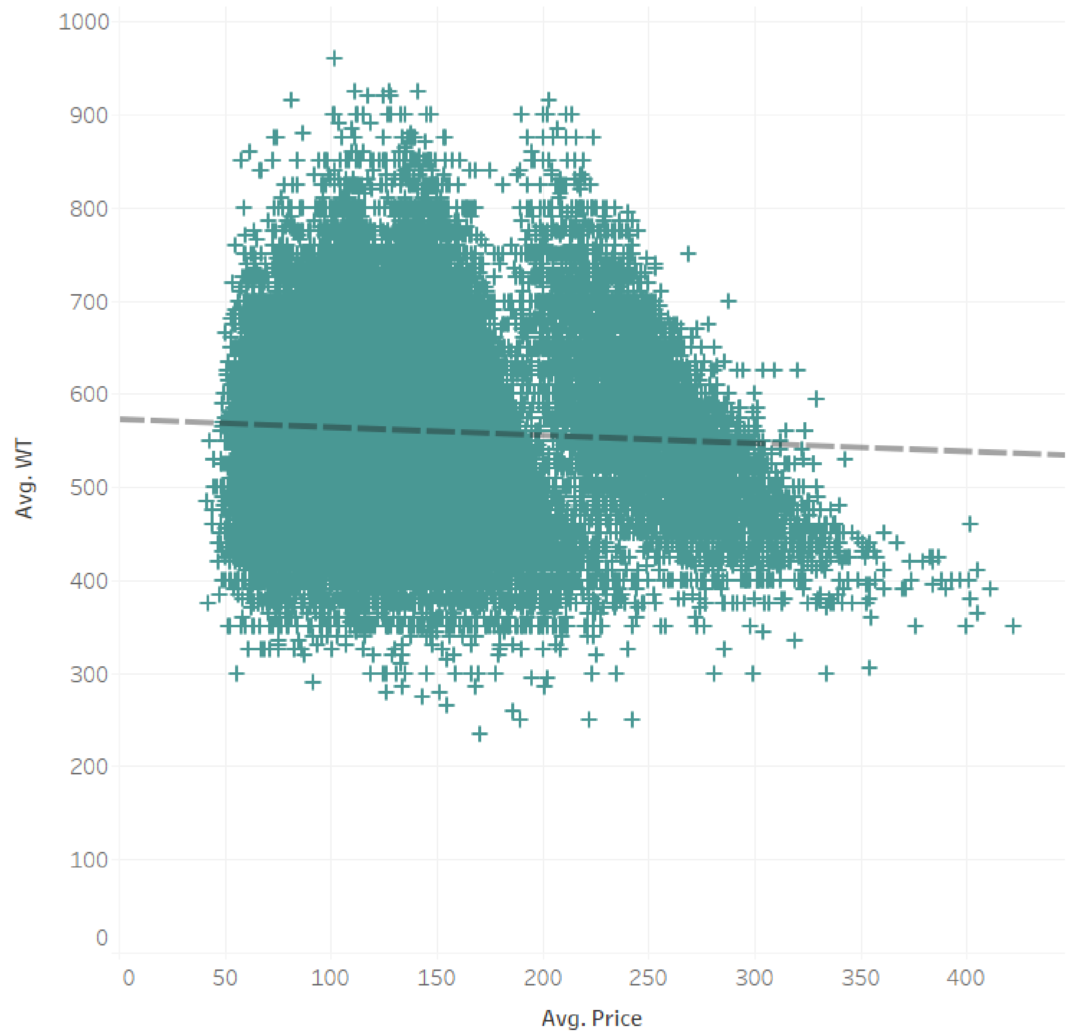
Average Head by Price



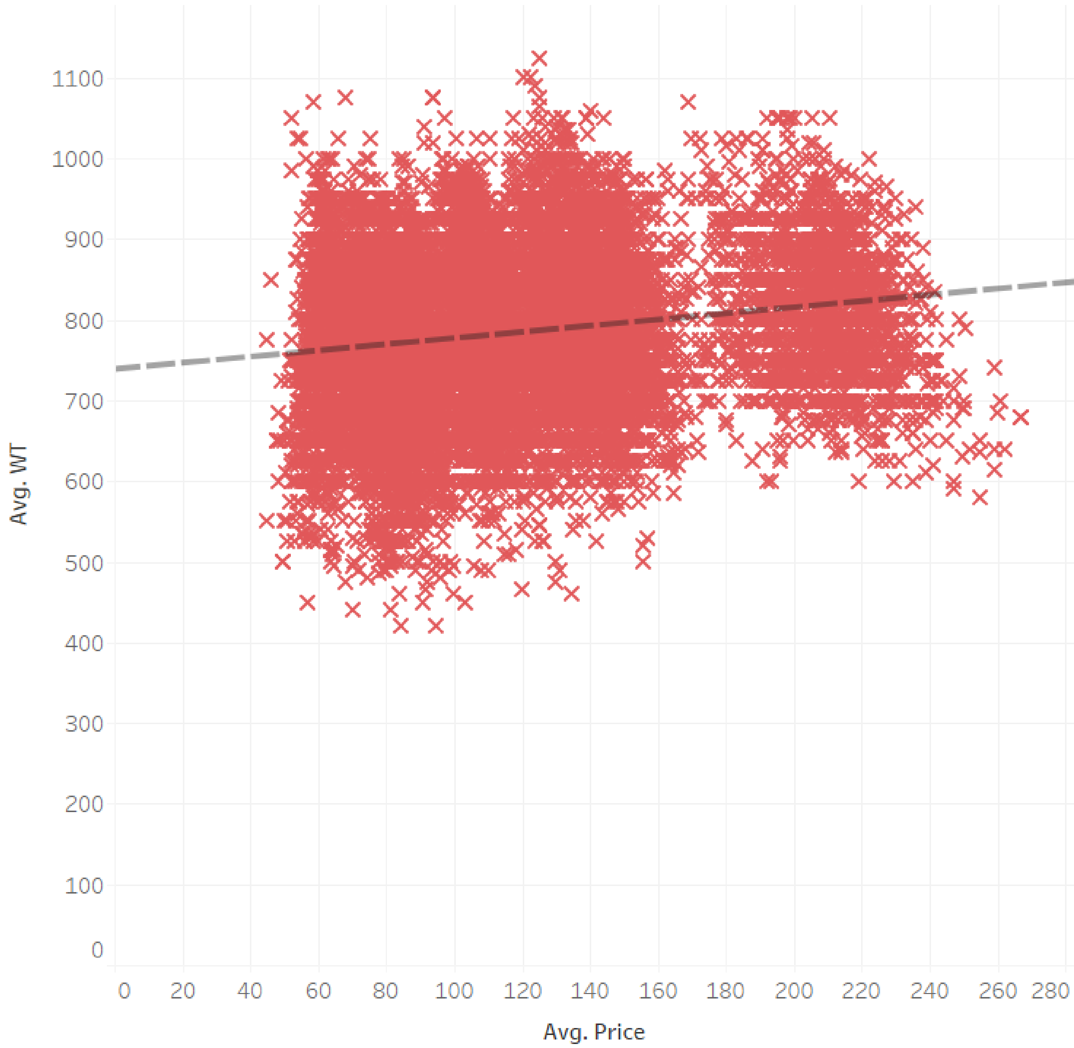


Average Weight by Price

Calves



Feeders



Model

Model

- SAS 9.4
- Multiple Regression
  - Backwards selection
  - $P < 0.05$
- Y = Sale Price
- X = Sale Year, Sex, Sale Area, Breed, Frame, Flesh, Vaccination, Weight, Size of lot
- Random = Sale Month (Sale Year)

Random = Sale Month (Sale Year)

Beef calf lots

Effect	SYEAR	Estimate	Pr > |t|
SMONTH(SYEAR)	1995	-1.2735	<.0001
SMONTH(SYEAR)	1996	0.1422	0.5216
SMONTH(SYEAR)	1997	-1.5064	<.0001
SMONTH(SYEAR)	1998	-0.8703	<.0001
SMONTH(SYEAR)	1999	0.7266	<.0001
SMONTH(SYEAR)	2000	-0.42	<.0001
SMONTH(SYEAR)	2001	-1.941	<.0001
SMONTH(SYEAR)	2002	0.2041	0.1665
SMONTH(SYEAR)	2003	2.6456	<.0001
SMONTH(SYEAR)	2004	1.3014	<.0001
SMONTH(SYEAR)	2005	-0.01458	0.9236
SMONTH(SYEAR)	2006	1.7367	<.0001
SMONTH(SYEAR)	2007	1.2434	<.0001
SMONTH(SYEAR)	2008	-1.1658	<.0001
SMONTH(SYEAR)	2009	-1.0244	<.0001
SMONTH(SYEAR)	2010	1.1077	<.0001
SMONTH(SYEAR)	2011	-0.7144	<.0001
SMONTH(SYEAR)	2012	-4.0628	<.0001
SMONTH(SYEAR)	2013	5.7739	<.0001
SMONTH(SYEAR)	2014	11.7076	<.0001
SMONTH(SYEAR)	2015	-12.5981	<.0001
SMONTH(SYEAR)	2016	-1.7284	<.0001
SMONTH(SYEAR)	2017	-1.011	<.0001
SMONTH(SYEAR)	2018	0.3321	0.045

Random = Sale Month (Sale Year)

Feeder cattle lots

Effect	SYEAR	Estimate	Pr > |t|
SMONTH(SYEAR)	1996	0.0372	0.834
SMONTH(SYEAR)	1997	-1.0354	<.0001
SMONTH(SYEAR)	1998	-0.5117	0.0008
SMONTH(SYEAR)	1999	0.787	<.0001
SMONTH(SYEAR)	2000	0.01985	0.7849
SMONTH(SYEAR)	2001	-1.136	<.0001
SMONTH(SYEAR)	2002	0.9689	<.0001
SMONTH(SYEAR)	2003	3.0202	<.0001
SMONTH(SYEAR)	2004	1.3053	<.0001
SMONTH(SYEAR)	2005	0.1485	0.2896
SMONTH(SYEAR)	2010	0.1651	0.2502
SMONTH(SYEAR)	2011	0.9521	<.0001
SMONTH(SYEAR)	2012	-1.9697	<.0001
SMONTH(SYEAR)	2013	5.4153	<.0001
SMONTH(SYEAR)	2014	7.0232	<.0001
SMONTH(SYEAR)	2015	-9.4307	<.0001
SMONTH(SYEAR)	2016	-1.9886	<.0001
SMONTH(SYEAR)	2017	-2.012	<.0001
SMONTH(SYEAR)	2018	1.7471	<.0001

Fixed Effects

Beef calf lots

Effect	Num DF	Pr > F
SYEAR	23	<.0001
WT	1	<.0001
SEX	1	<.0001
SAREA	3	<.0001
BREED	2	<.0001
FRAME	2	<.0001
FLESH	3	<.0001
VAC	1	<.0001
HEAD	1	<.0001

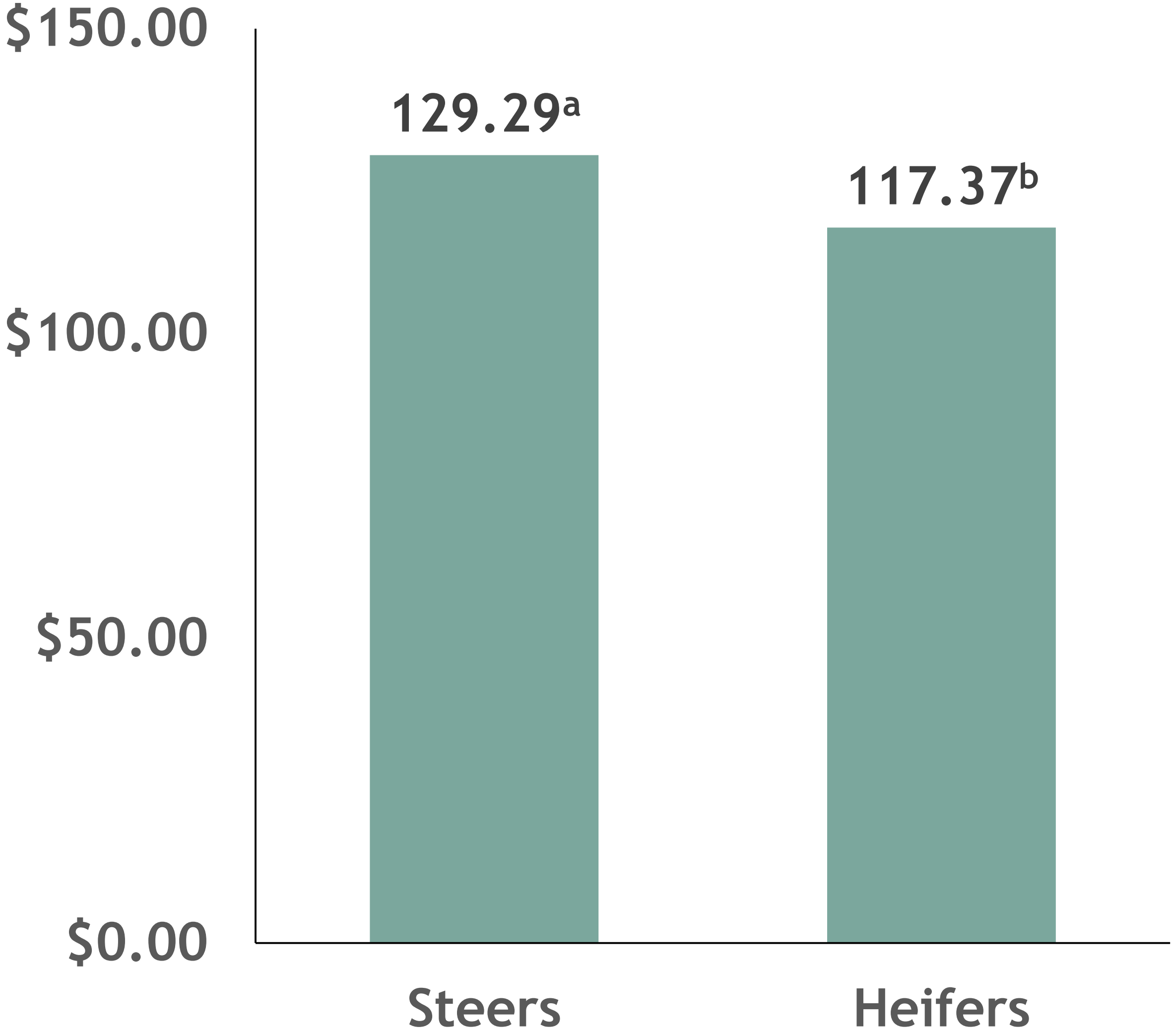
Feeder cattle lots

Effect	Num DF	Pr > F
SYEAR	18	<.0001
WT	1	<.0001
SEX	1	<.0001
SAREA	3	<.0001
BREED	2	<.0001
FRAME	2	<.0001
FLESH	3	<.0001
HEAD	1	<.0001


Sale Price by Year
Beef calf lots
$350.00
$300.00
$250.00
$200.00
$150.00
$100.00
$50.00
$0.00
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Feeder cattle lots
$350.00
$300.00
$250.00
$200.00
$150.00
$100.00
$50.00
$0.00
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2010
2011
2012
2013
2014
2015
2016
2017
2018

Sale Price by Gender

Beef calf lots

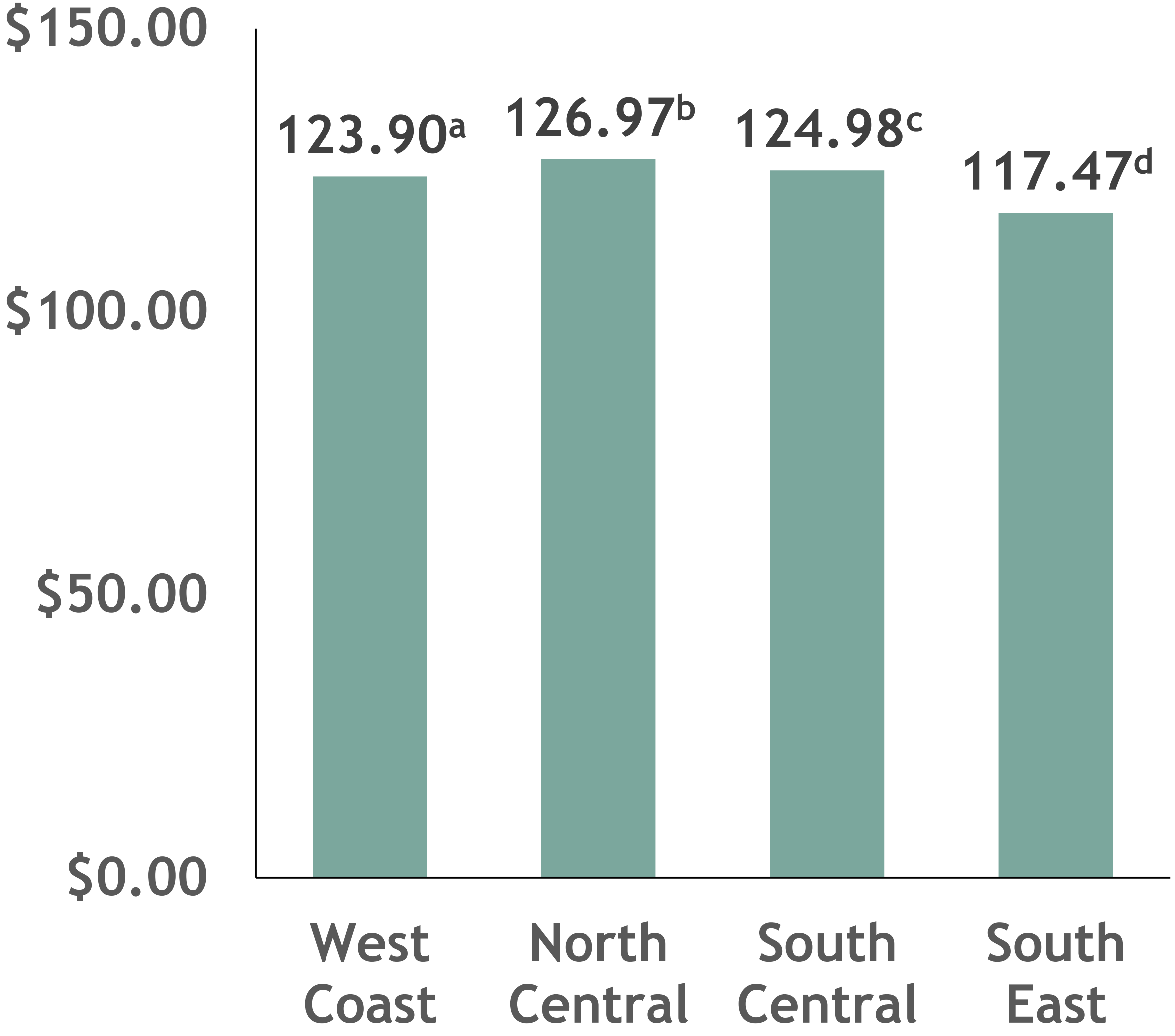


Feeder cattle lots

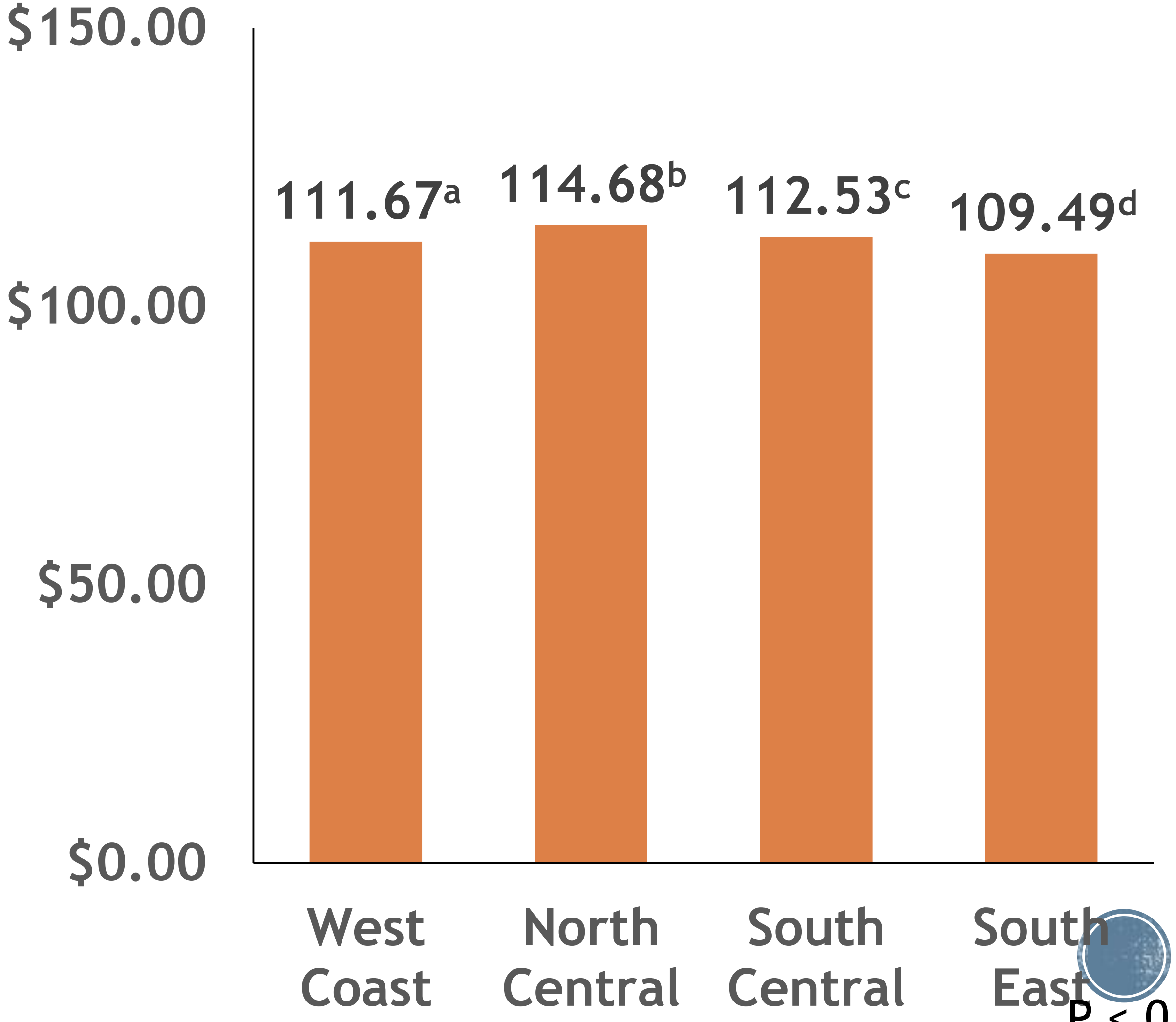
$P < 0.05$

Sale Price by Sale Area

Beef calf lots

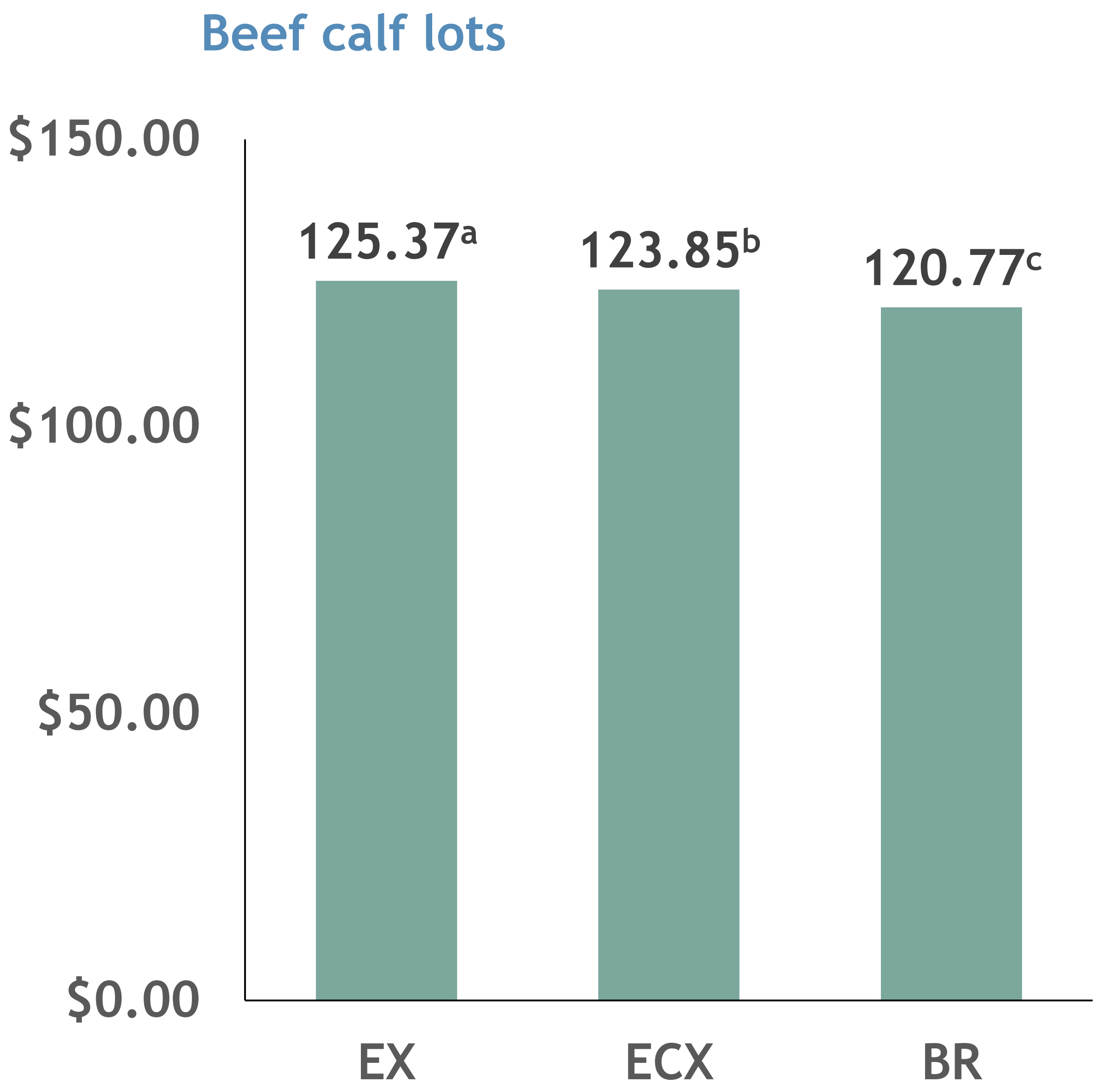


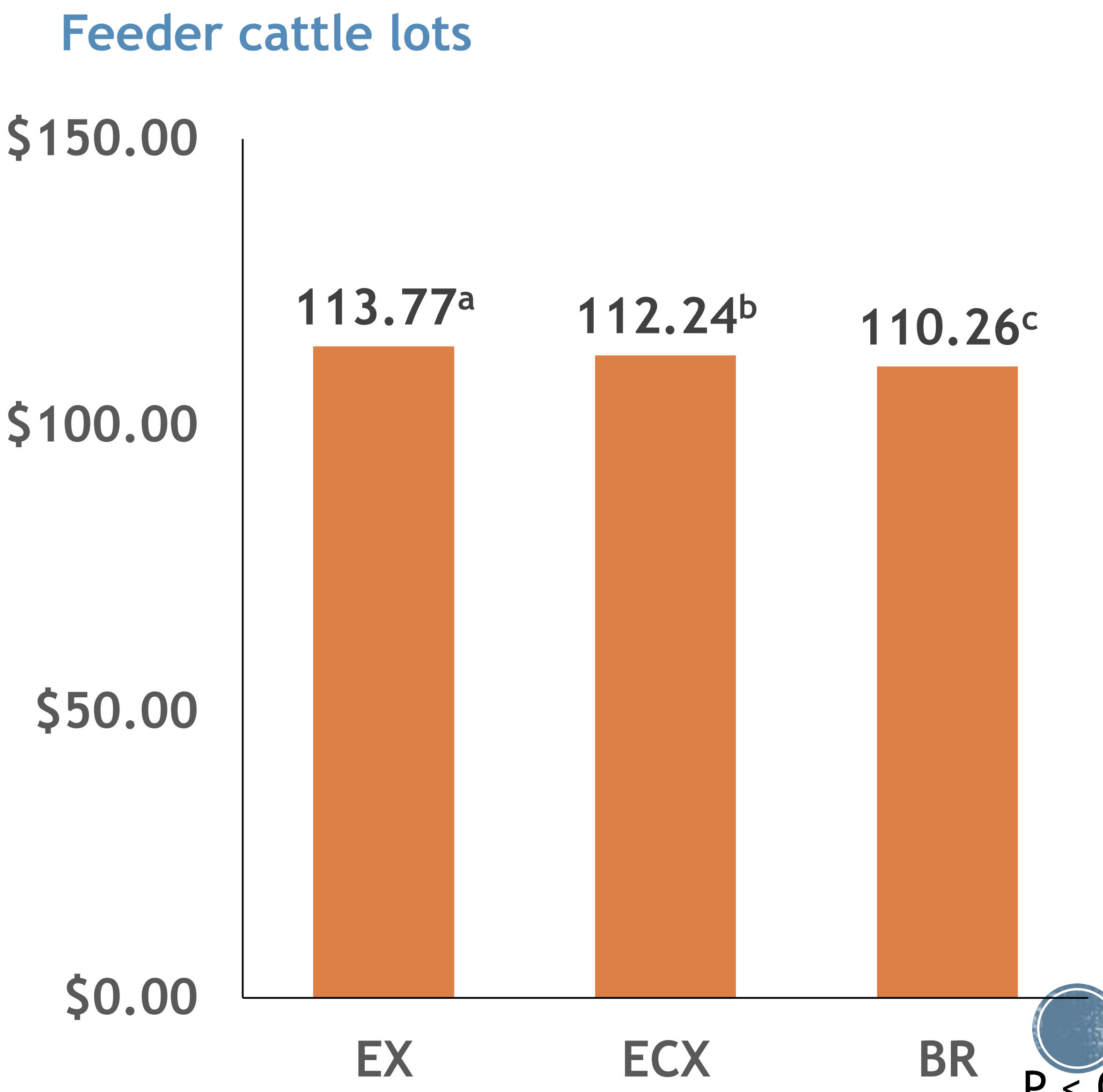
Feeder cattle lots



$P < 0.05$

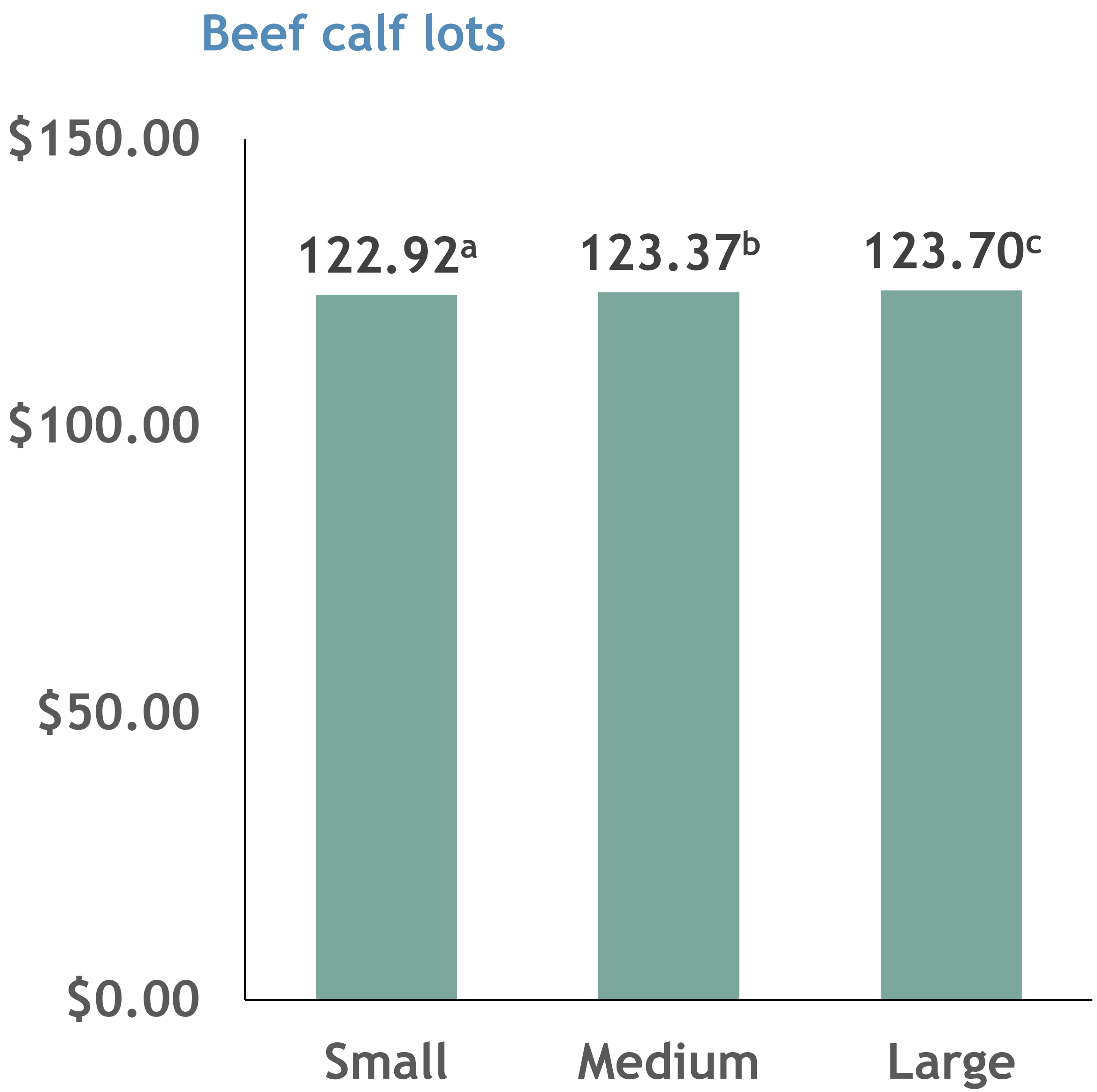
Sale Price by Breed

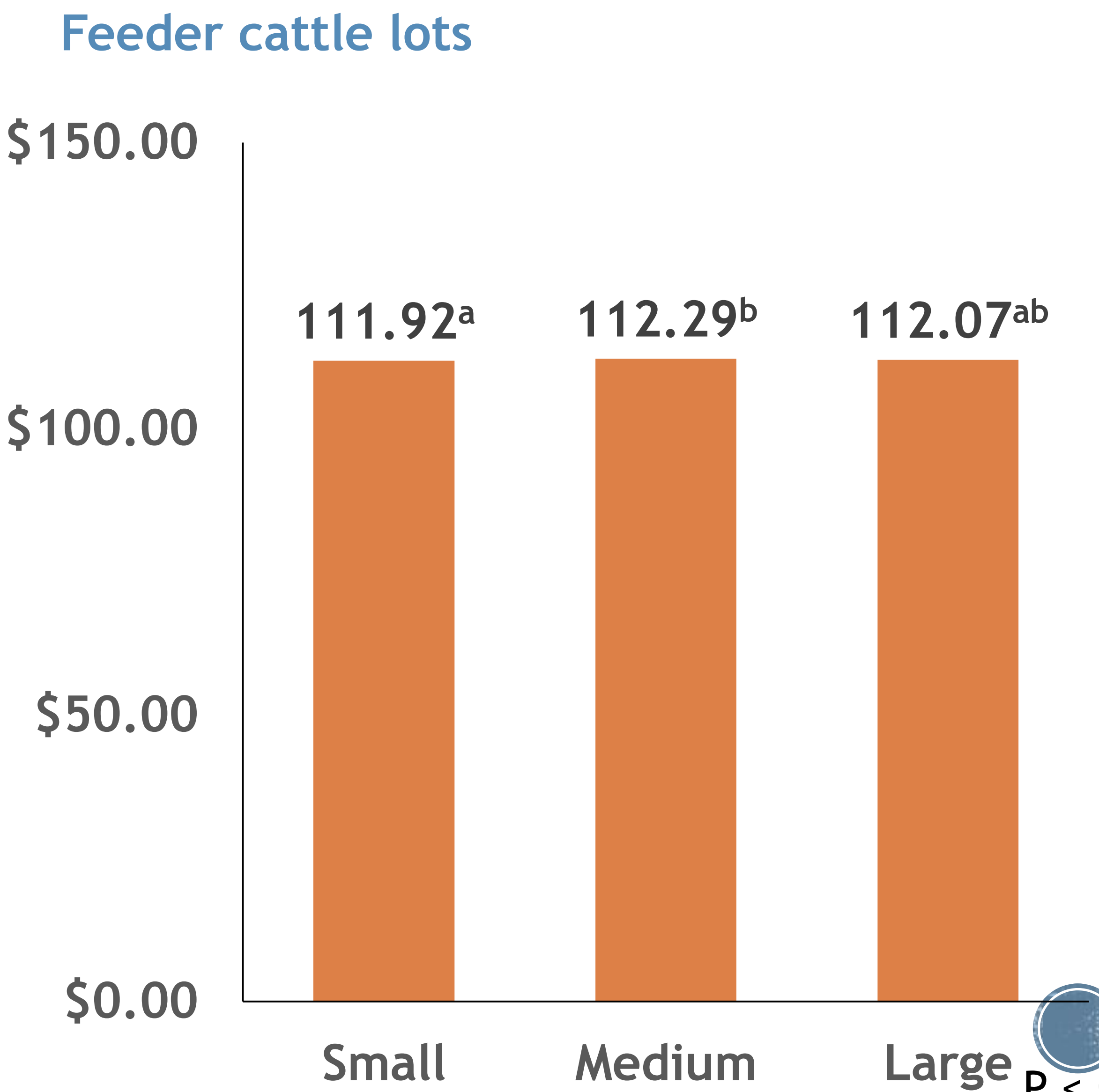




$P < 0.05$

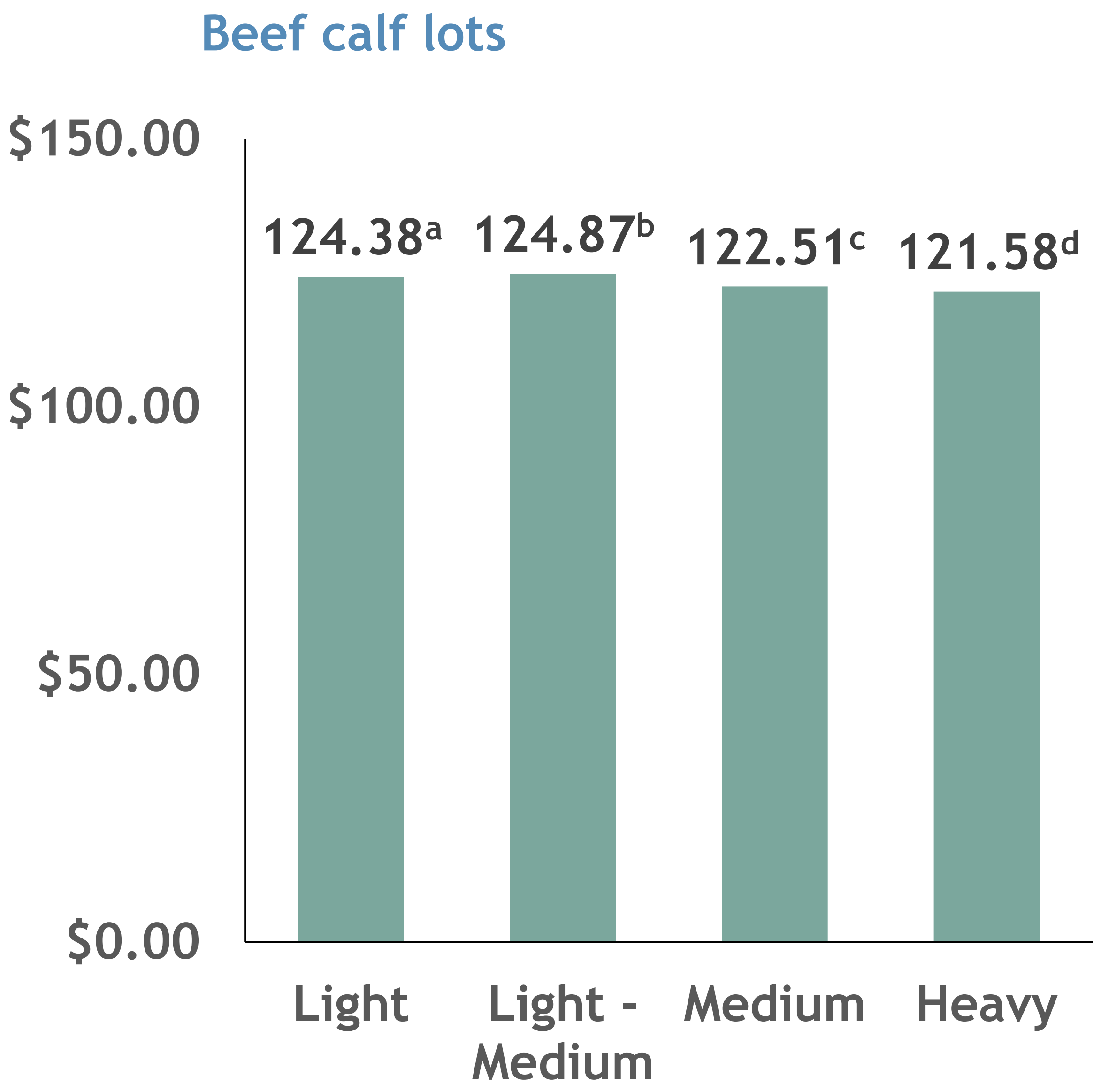
Sale Price by Frame

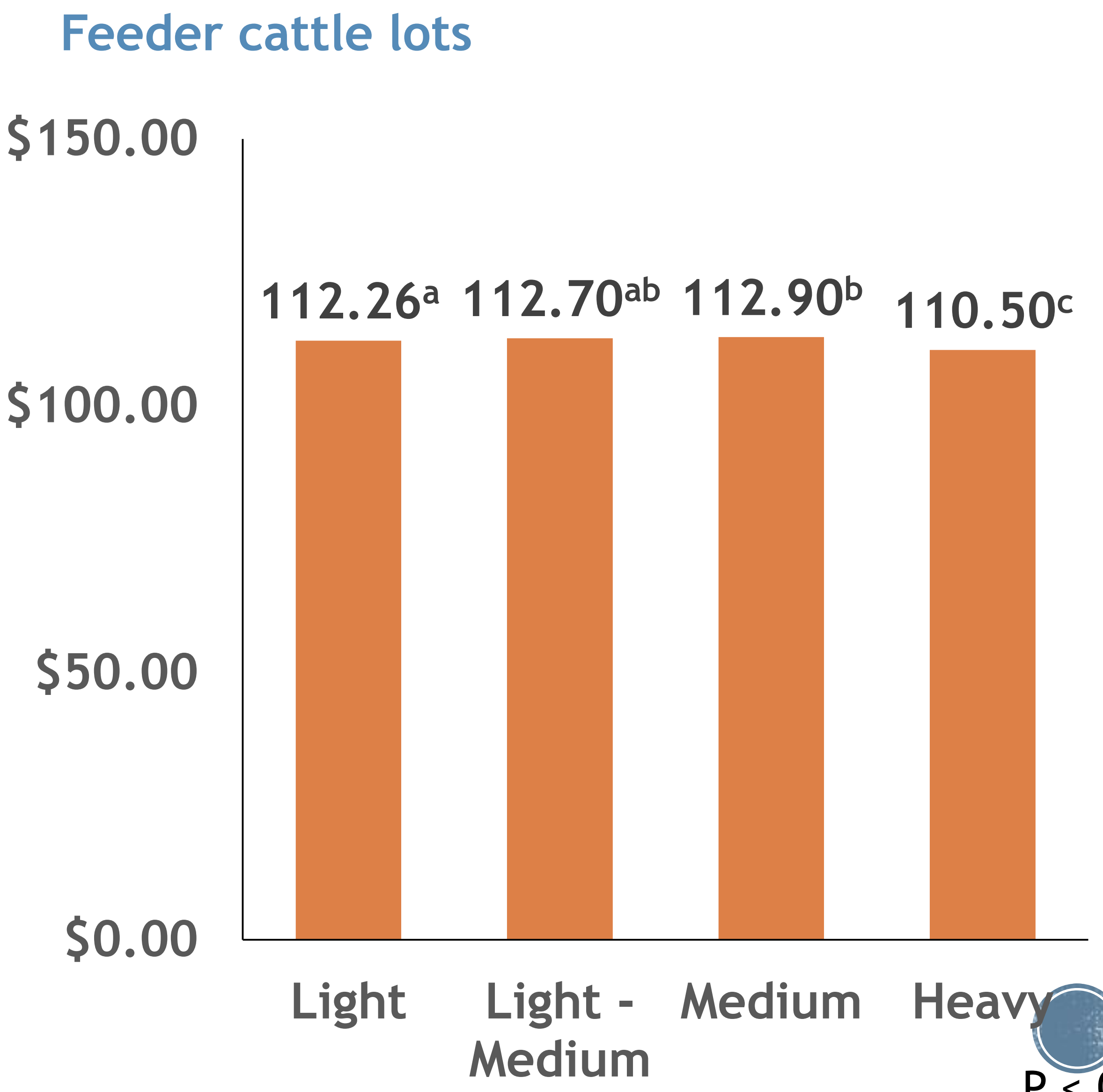




$P < 0.05$

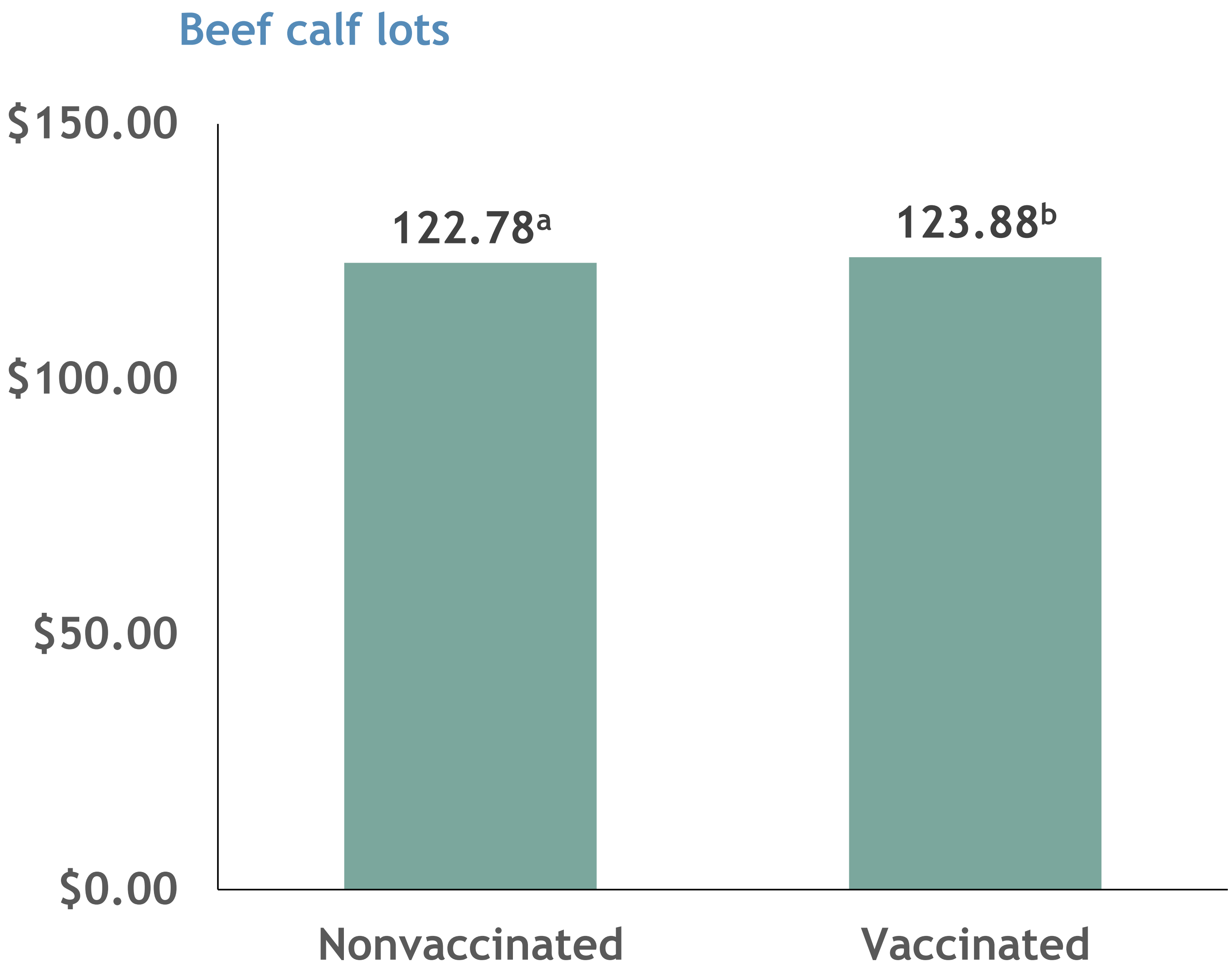
Sale Price by Flesh





$P < 0.05$

Sale Price by Vaccination Status



P < 0.05

Model Validation

Model Validation

In [913]:
```
# import dataset
SLA1=pd.read_csv("DATA/SLA.csv")
SLA1.head()
```

Out[913]:

	Unnamed: 0	SYEAR	LOTID	SMONTH	HEAD	SEX	WT	PRICE	STATE	STATECODE	SAREA	BREED	FRAME	FLESH	VAC
0	0	1995	1	6	115	1	425	81.25	OK	25	3	3	4	4	1
1	1	1995	2	6	220	1	460	80.00	OK	25	3	3	4	4	1
2	2	1995	3	6	84	1	570	77.00	MO	23	3	1	4	4	1
3	3	1995	4	6	83	1	610	74.50	KS	22	3	3	5	4	0
4	4	1995	5	6	155	1	625	68.10	MS	46	5	4	3	4	0

In [914]:
```
SLA1 = SLA1.drop(['Unnamed: 0', 'LOTID', 'STATE', 'STATECODE'], axis=1)
SLA1.head()
```

Out[914]:

	SYEAR	SMONTH	HEAD	SEX	WT	PRICE	SAREA	BREED	FRAME	FLESH	VAC
0	1995	6	115	1	425	81.25	3	3	4	4	1
1	1995	6	220	1	460	80.00	3	3	4	4	1
2	1995	6	84	1	570	77.00	3	1	4	4	1
3	1995	6	83	1	610	74.50	3	3	5	4	0
4	1995	6	155	1	625	68.10	5	4	3	4	0

In [915]:
```
SLA1 = SLA1.astype({"PRICE": int, "WT": int, "HEAD":int})
```

In [916]:
```
y = SLA1['PRICE']
X = SLA1[['SYEAR','SAREA','BREED','WT','HEAD','FRAME','SEX', 'VAC']]
```

In [917]:
```
# create training and testing vars
X_train, X_test, y_train, y_test = train_test_split(SLA1, y, test_size=0.3)
print X_train.shape, y_train.shape
print X_test.shape, y_test.shape
```

```
(66410, 11) (66410L,)
(28462, 11) (28462L,)
```

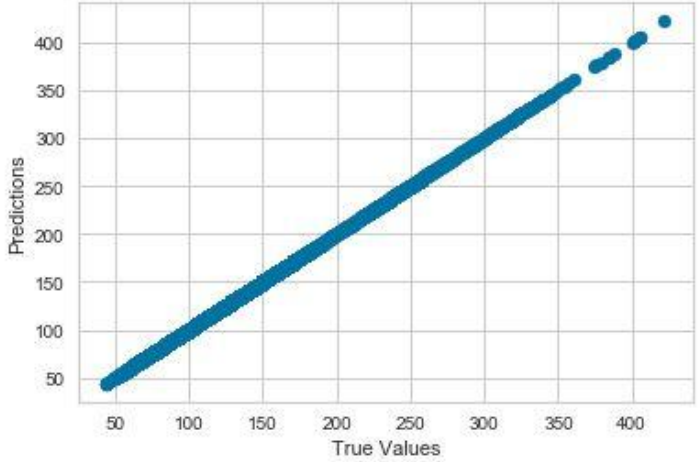
```
In [918]: # fit a model
          lm = linear_model.LinearRegression()
          model = lm.fit(X_train, y_train)
          predictions = lm.predict(X_test)

In [919]: predictions[0:5]

Out[919]: array([ 74., 125., 173., 158., 226.])

In [920]: ## The line / model
          plt.scatter(y_test, predictions)
          plt.xlabel('True Values')
          plt.ylabel('Predictions')

Out[920]: Text(0,0.5,'Predictions')
```




```
In [921]: print 'Score:', model.score(X_test, y_test)

          Score: 1.0

In [922]: # evaluate the decision tree model using 10-fold cross-validation
          scores = cross_val_score(dt, X, y, scoring='accuracy', cv=10)
          print scores
          print scores.mean()

          [0.03292266 0.04167969 0.04877028 0.04546886 0.04213187 0.03835182
           0.04071247 0.04526618 0.04626372 0.04061999]
          0.042218755176322675
```

In [923]:
```
# split validation
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Initialize DecisionTreeClassifier()
dt = DecisionTreeClassifier()

# Train a decision tree model
dt.fit(X_train, y_train)
```

Out[923]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [924]:
```
#Model evaluation
# http://scikit-learn.org/stable/modules/model_evaluation.html
print metrics.accuracy_score(y_test, dt.predict(X_test))
print "--------------------------------------------------------"
print metrics.confusion_matrix(y_test, dt.predict(X_test))
print "--------------------------------------------------------"
print metrics.classification_report(y_test, dt.predict(X_test))
print "--------------------------------------------------------"
#print metrics.roc_auc_score(y_test, dt.predict(X_test))

# y-test is the acual y value in the testing dataset
# dt.predict(X_test) is the predicted y value generated by your model
# If they are same, we can say your model is accurate.
```

```
0.0730961791831357
--------------------------------------------------------
[[0 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
--------------------------------------------------------
             precision    recall  f1-score   support

         41       0.00      0.00      0.00         0
         44       0.67      0.67      0.67         3
         46       0.00      0.00      0.00         0
```

Model Validation

In [997]:
```
# import dataset
FEEDER1=pd.read_csv("DATA/FEEDER1.csv")
FEEDER1.head()
```

Out[997]:

	Unnamed: 0	SYEAR	LOTID	SMONTH	HEAD	SEX	WT	PRICE	STATE	STATECODE	SAREA	BREED	FRAME	FLESH	VAC
0	0	2010	111922	6	39	1	640	105.25	TX	26	3	4	4	4	0
1	1	2010	111923	6	39	2	640	98.25	TX	26	3	4	4	4	0
2	2	2010	111924	6	35	1	700	105.50	LA	45	5	4	3	2	0
3	3	2010	111925	6	35	2	700	99.50	LA	45	5	4	3	2	0
4	4	2010	111926	6	32	1	735	105.25	TX	26	3	4	4	4	1

In [998]:
```
#now we can drop the orgional columns we created the dummy variables for
FEEDER1 = FEEDER1.drop(['Unnamed: 0', 'LOTID', 'STATE', 'STATECODE', 'VAC'], axis=1)
FEEDER1.head()
```

Out[998]:

	SYEAR	SMONTH	HEAD	SEX	WT	PRICE	SAREA	BREED	FRAME	FLESH
0	2010	6	39	1	640	105.25	3	4	4	4
1	2010	6	39	2	640	98.25	3	4	4	4
2	2010	6	35	1	700	105.50	5	4	3	2
3	2010	6	35	2	700	99.50	5	4	3	2
4	2010	6	32	1	735	105.25	3	4	4	4

In [999]:
```
FEEDER1 = FEEDER1.astype({"PRICE": int, "WT": int, "HEAD":int})
```

In [1000]:
```
y = FEEDER1['PRICE']
X = FEEDER1[['SYEAR','SAREA','BREED','WT','HEAD','FRAME','SEX']]
```

In [1001]:
```
# create training and testing vars
X_train, X_test, y_train, y_test = train_test_split(FEEDER1, y, test_size=0.3)
print X_train.shape, y_train.shape
print X_test.shape, y_test.shape
```

```
(16703, 10) (16703L,)
(7159, 10) (7159L,)
```

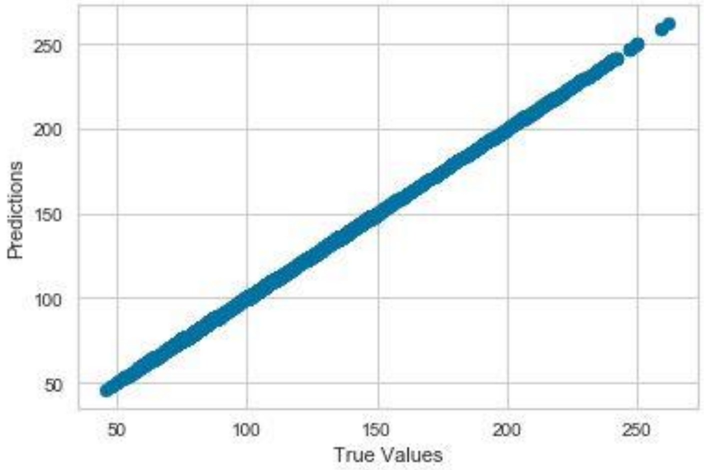
In [1002]:
```
# fit a model
lm = linear_model.LinearRegression()
model = lm.fit(X_train, y_train)
predictions = lm.predict(X_test)
```

In [1003]:
```
predictions[0:5]
```

Out[1003]:
```
array([139.,  74., 131., 110.,  62.])
```

In [1004]:
```
## The line / model
plt.scatter(y_test, predictions)
plt.xlabel('True Values')
plt.ylabel('Predictions')
```

Out[1004]:
```
Text(0,0.5,'Predictions')
```



In [1005]:
```
print 'Score:', model.score(X_test, y_test)
```

```
Score: 1.0
```

In [1006]:
```
# evaluate the decision tree model using 10-fold cross-validation
scores = cross_val_score(dt, X, y, scoring='accuracy', cv=10)
print scores
print scores.mean()
```

```
[0.07157218 0.07832792 0.07216072 0.07281754 0.06892231 0.08185654
 0.0625     0.07596567 0.06358131 0.05790161]
0.07056057981651372
```

In [1007]:
```
# split validation
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Initialize DecisionTreeClassifier()
dt = DecisionTreeClassifier()

# Train a decision tree model
dt.fit(X_train, y_train)
```

Out[1007]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [1008]:
```
#Model evaluation
# http://scikit-learn.org/stable/modules/model_evaluation.html
print metrics.accuracy_score(y_test, dt.predict(X_test))
print "--------------------------------------------------------"
print metrics.confusion_matrix(y_test, dt.predict(X_test))
print "--------------------------------------------------------"
print metrics.classification_report(y_test, dt.predict(X_test))
print "--------------------------------------------------------"
#print metrics.roc_auc_score(y_test, dt.predict(X_test))

# y-test is the acual y value in the testing dataset
# dt.predict(X_test) is the predicted y value generated by your model
# If they are same, we can say your model is accurate.
```

```
0.1169076052796983
--------------------------------------------------------
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
```

Managerial and Technical Implications

Implications

- Steers sell for more than heifers
  - Why?
    - More efficient in feed conversion, less management and risk (i.e. pregnancy)
- Cattle located near the Midwest have the highest sale prices
  - Why?
    - Located closer to where the feedyard and resources are located to finish cattle, less transportation cost/risk
- What breed of cattle have the most value?
  - English type cattle (i.e. Angus, Hereford)
  - English-Continental crossed cattle (i.e. Angus X Charolais)
  - Brahman-influenced cattle have the lowest sale price
- Larger frame, light/medium flesh
- Vaccinated

Implications

- Don't "over trust" the data
- Observational, producer data
  - Limitations
    - Why will the buyer value?
    - What is practical for each producer?
- Representative cowherds of +300 head
  - Average herd size in United States is 40 head
- Strive for premiums or avoid discounts?
  - Market changes impact demand

Questions?